AMIGA WELT

2/89 März/April DM 12,–



Flugsimulatoren

Absturz ohne Risiko

ISSN 0933-8616

Neuheiten zur CeBIT

GFA-BASIC im TEST

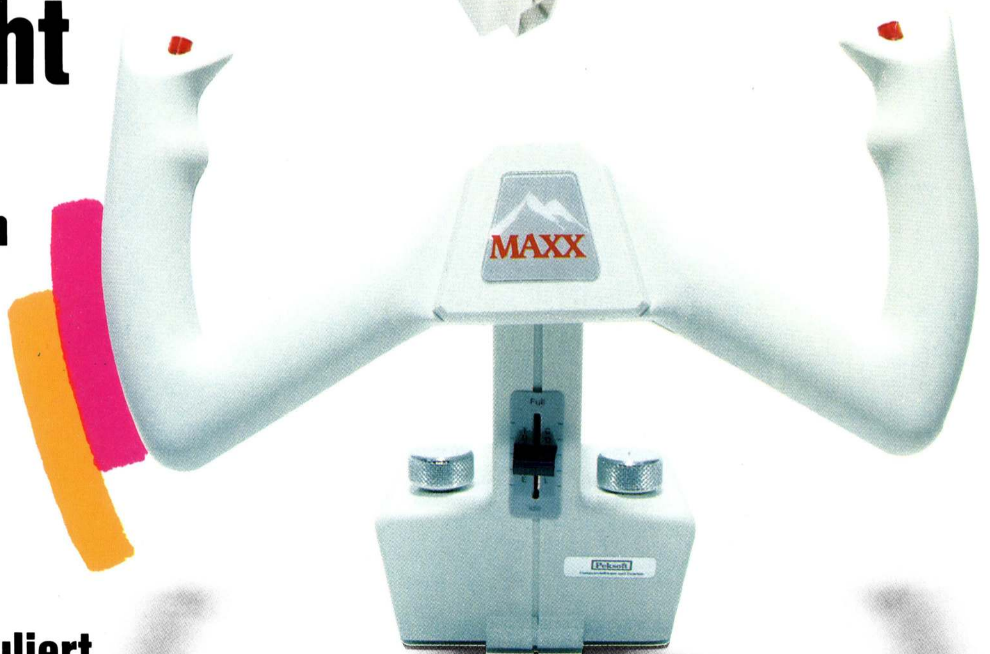
Man spricht Amiga

Programmiersprachen im Vergleich

Sammeln!

10 Seiten deutsche Anleitungen für PD

Superlisting: Zaubermeister manipuliert Grafik



sfr. 12,—; ÖS 95,—; Fmk 41,—; hfl 15,—; Lire 12500; lfr. 300;

Tastaturschablonen: FLIGHT II für 500er, 1000er und 2000er | DPaint für 1000er

Profilaufwerk 3,5"
Metallgehäuse ● einstellbare Lauwerknummer mit Displayanzeige ● digitale Trackanzeige ● Write Protect am Laufwerk schaltbar ● abschaltbar ● durchgeschleifter Bus
1 Jahr Garantie
Super ALCOMPreis **329,-**

Laufwerk 5,25"
40/80 Track ● Laufwerksbus durchgeschleift ● abschaltbar ● einstellbare Adressen ● MS- DOS- kompatibel ● mit Diskchange

Super ALCOMPreis	298,-
HD 1,6 MB (umschaltbar)	318,-
Amigafarbene Blende	+10,-
Write Protect Schalter	+15,-

Gemischtes Doppel 3,5/5,25"
einzeln ein-/abschaltbar ● einstellbare Laufwerksnummern mit Anzeige ● durchgeschleifter Bus ● bei 5,25" 40/80 Tracks umschaltbar ● Metallgehäuse ● 1 Jahr Garantie
Super ALCOMPreis **548,-**

500er Speichererweiterung
Für 512k zusätzliches RAM ● alle RAM-s gesockelt ● selbstkonfigurierend ● abschaltbar ● Uhrenschaltung auf Platine mit Akku- bzw. Batteriepufferung nachrüstbar

Komplett mit 512k	**Preis auf Anfrage**
Superpreis mit Uhr	**Preis auf Anfrage**
Bauteilesatz für Uhr ohne Akku	24,-
Leerplatine mit Stecker	*39,-

*mit Schaltplan und Bestückungsliste

Laufwerkanschlußkabel
Zum Anschluß von Laufwerken an alle Amigas ● mit Ansteuerelektronik

Für 3,5" Laufwerk	39,-
Für 5,25" Laufwerk	49,-

Steckplatzerweiterung 3-fach für Laufwerke
Jeder Steckplatz abschaltbar und einstellbare Laufwerksnummer ● Steckplatzerweiterung direkt am Amigagehäuse ● Dadurch keine Kabellängenprobleme
Anschlußfertig zum Super ALCOMPreis **49,-**

Soundsampler
Für alle Amiga's mit Software ● Type bei Bestellung bitte angeben ● 8-Bit Datenbreite ● Betrieb am Parallelport (Druckerport) ● Mit Vorverstärker für Micro-Anschluß (Cinch-Buchsen) ● Musik- und Sprachdigitalisierung möglich ● Arbeitet mit fast allen Digitizer-Programmen ● Formschönes Gehäuse
Super ALCOMPreis **79,-**

Sampler Studio
● Professionelles Sampler-Programm ● 4-Kanal-Technik ● speichern auf 4 Disketten hintereinander möglich ● alle gängigen Formate (IFF, Data, Future) ● Echtzeitdisplay mit Zoomfunktion ● viele Verfremdungsmöglichkeiten ● Echo, Hall, Reverse
69,-
Paket: Sampler + Software **129,-**

MIDI - Interface
4 Kanäle einschließlich 1 Thru ● Optische Datenanzeige ● Formschönes Gehäuse
Wahnsinnspreis von nur **89,-**

Frischer Wind auf dem Hardwaresektor

ALCOMP
COMPUTERHARDWARE

Kickstartumschaltung
Bauen Sie die anderen Kickstart-Versionen in Ihren Amiga 500 ● Einfacher Einbau ohne Löten ● für Original-Kickstart-ROM und 2 zusätzliche Versionen auf EPROM ● EPROM-Programmierservice auf Anfrage

SuperALCOMPreis	59.-
Kickstartversion auf EPROM's	120.-

Userport + Experimentierkarte für Expansionport
Mit Lochraster und 2 x 6522 Ports

Leer	59.-
komplett aufgebaut	89.-

Wir suchen ständig Hardware-Entwicklungen. Wir garantieren gute Umsatzprovisionen und ehrliche Abrechnung

ausgereifte Ingenieurleistung ● 14 Tage Umtauschrecht ● fast alle IC's gesockelt ● nur professionelle Leiterplatten ● Bauteile namhafter Hersteller ● mit Bedienungsanleitung



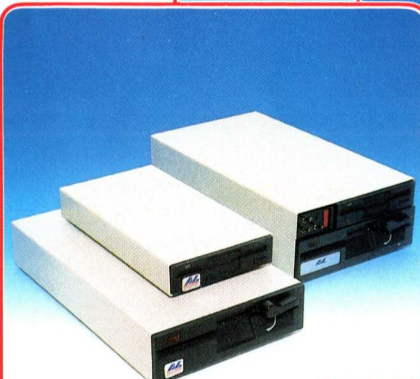
kostenloses Info anfordern!!!

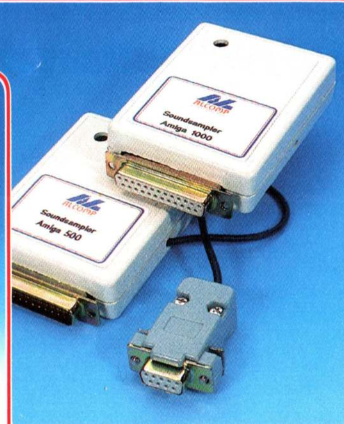
Bestellung und Versand
**ALCOMP GmbH
Glescher Weg 22
5012 Bedburg
Tel. 0 22 72/20 93**

Nachnahmeversand NN-Spesen 7.50 DM b. Vorkasse 3.- DM. Auslandsbestellungen: Nachnahmeversand NN-Spesen 10.- DM b. Vorkasse 5.-DM. Wir liefern Ihnen auf Ihre Rechnung und Gefahr zu den Verkaufs- und Lieferbedingungen des Elektronikgewerbes. Postgiroamt Köln (BLZ 370 100 50) 275 54-509

3,5" Laufwerk
Für alle Amiga's ● einstellbare Gerätenummer ● abschaltbar ● Metallgehäuse ● superflach ● 1 Zoll (2,54cm) ● durchgeschleifter Bus ● TEAC Laufwerk
1 Jahr Garantie

komplett anschlußfertig	239,-
Amigafarbene Blende	+10,-





Basislaufwerke
1 Jahr Garantie

TEAC FD 135 FN 3,5" 1MB superslimline	218,-
1,6 MB Diskchange	
Amigafarbene Blende	+10,-
3,5" Gehäuse	25,-
5,25" Gehäuse	25,-
Gehäuse für "Gemischtes Doppel"	65,-

Bootselector
19,90

Amiga Eprommer
- Für A 500/1000
- Expansionsportanschluß
- Für EPROM's 2764-27011 (8K-128K)
  Alle A-Typen und CMOS-Typen
- Funktionen:
  LEERTEST — LADEN VON DISK
  VERGLEICHEN — SPEICHERN AUS DISK
  AUSLESEN — HEXDUMP
  BRENNEN
- vier Programmieralgorithmen
  50mS/Byte - Superschnell 64K-1,5 min
- Programm zum Generieren und Brennen von Kickstarts direkt von Diskette oder aus ROM
- Mit Software + Gehäuse **225,-**

Meß- und Steuerinterface
- 8 ADC-Kanäle 0-2,55V in 0,01V Stufe
- 1 DAC-Kanäle 0-2,55V in 0,01V Stufe
  Genauigkeit- 1,5 LSB
- 8 frei programmierbare TTL-I/O Kanäle
- Mit Gehäuse, Anschlüße auf Schraubklemmen
- interne Referenzspannung
- Expansionsanschluß
- Einfache Programmierung in Basic möglich Multitasking tauglich
- incl. DEMO-Software auf 3,5" Diskette

239,-

Trackanzeige
Für DF0-DF3 einstellbar ● für alle Laufwerke (3,5"/5,25") ● Laufwerkbus durchgeschleift ● mit Gehäuse
Super ALCOMPreis **69,-**

Einführungsangebot Amiga - Harddisks
komplett anschlußfertig

Platte 20 MB A 2000	798,-
30 MB A 2000	898,-
40 MB A 2000	1 098,-
65 MB A 2000	1 348,-
Platte A 500/A 1000	
20 MB	898,-
30 MB	998,-
40 MB	1 248,-
65 MB	1 498,-

für den Selbstbau

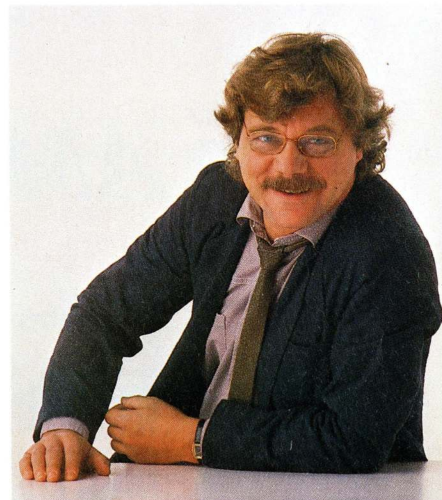
HD-Interface A 2000	198,-
HD-Interface A 500/A 1000	249,-

Vokabeltrainer
2500 englisch-deutsche Vokabeln incl. Hilfssatz ● Merkfunktion ● komfortabler Editor zur Vokabelverwaltung ● Wörterbuch zum Dateiendurchsuchen
59,-



Selbstbootende Harddisk für Amiga ohne PC-Karte!
Die Amiga-Festplatte von ALCOMP:
● Selbstbootend wie "Card" oder "Rad"! ● Als Einbau-Festplatte für den "Amiga 2000" ● Als Externe Einheit für den "Amiga 500" und 1000 mit Gehäuse, eigenem Netzteil und Erweiterungsanschluß ● Erhältlich mit 20, 30, 40 und 65 Megabyte ● Kopiert 1 Megabyte in unter 4 Sekunden ● Speichert schneller als "1.2-Ramdisk" ● Läuft mit "FastFileSystem" ● Einfach einstecken, Formatieren, "Mountlist" und "Startup-Sequence" ändern und los geht's!
Entwickler: Stephan und Stefan
Für den Selbstbau: Harddisk-Interface incl. Steuersoftware ● Anschluß mit Slot für Omti-Controller

Entwurf + Druck PLUM, Düren-Oberzier, TEL. 02428/711


Manfred S. Schmidt (Chefredakteur)

Apropos ... Crime and Sex

In ein Türkentaxi muß man Handgranaten hineinwerfen! Diese kriminelle Handlungsanweisung existiert nicht erst seit den Berliner Wahlen, sondern ist Bestandteil eines von der Bundesprüfstelle für jugendgefährdende Schriften (BPS) indizierten Computerprogrammes. Da wird der Posten eines Berliner Gauleiters zum erstrebenswerten Traumberuf, Juden sind die dümmsten Tiere und Deutschland hat den Zweiten Weltkrieg gewonnen. Der Senator für Jugend und Familie des Landes Berlin hat mit Erfolg beantragt, das Computerspiel in die Liste der jugendgefährdenden Schriften aufzunehmen. Zur Begründung führte der Senat an, das Programm sei besonders sittlich gefährdend. Das Programm nutze nur einen geringen Teil der Speicherkapazität einer Diskette und werde in den Kreisen der Raubkopierszene zusammen mit anderen Programmen getauscht und weiter vervielfältigt. Der Inhalt der Texte sei für Kinder und Jugendliche sittlich gefährdend, da er verrohend wirke und zum Rassenhaß anreize. Das, was ich beim besten Willen nicht mehr als Spiel bezeichnen will, besteht aus 21 Fragen mit je drei bis vier vorgegebenen möglichen Antworten, aus denen der Spieler eine auswählen soll, sowie vorgegebene Bewertungen der Antwort. Das Ganze beginnt mit der Präsentation des Hakenkreuzes und endet damit, daß das Programm sich „ordnungsgemäß" abmeldet. Politisch extremistisches Gedankengut stellt, und das soll hier nicht verschwiegen werden, Gott sei dank einen vergleichsweise geringen Teil des Gesamtindex dar. Bei der Mehrheit der indizierten Programme handelt es sich um Kriegsspiele oder Pornosoftware. Darüber läßt es sich diskutieren: ist die statische Abbildung eines kopulierenden Paares bereits jugendgefährdend? Hat die Animation zum Geschlechtsverkehr ähnlich schädliche Wirkungen wie die Aufforderung zum Totschlag?

Zweifel sind angebracht. Zweifelhaft ist aber nicht nur die Sex-and-Crime-Software an sich. Dubios ist auch das Geschäftsgebaren jener, die dieses Material herstellen bzw. vertreiben. Da werden Urheberrechte verletzt, indem Fotos aus Herrenmagazinen unerlaubt kopiert werden. Da werden Versprechungen gemacht (spannend, prickelnd, sexy, etc.) und nicht gehalten. Da wird sogenannte Adult-Software ohne weiteres an Minderjährige verschickt. Und es wird kassiert und gelegentlich nicht geliefert. Das doofe Opfer traut sich nicht, Beschwerde zu führen. Wir von der AmigaWelt wollen den Finger auf einen wunden Punkt legen. Wir möchten wissen, welche Erfahrungen unsere Leser in diesem Zusammenhang gemacht haben. Also bitte ich Sie, mit mir Kontakt aufzunehmen. Schreiben Sie mir oder rufen Sie mich an (Tel. 089/ 3 60 86-210), wenn es sein muß, auch anonym. Je ausführlicher und genauer, desto besser. Unter den Einsendern verlosen wir 10 Programmdisketten der AmigaWelt. Einsendeschluß ist der 3. März 1989.

Manfred S. Schmidt

Manfred S. Schmidt

INHALT

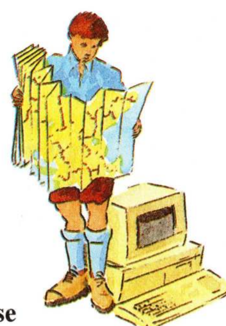
Software, die beflügelt: Flightsimulator II und III, Falcon F-16 und Scenary Disks. Komplexe Cockpit-Technik landet auf dem Bildschirm – der User hebt ab und schwebt in die Ferne.



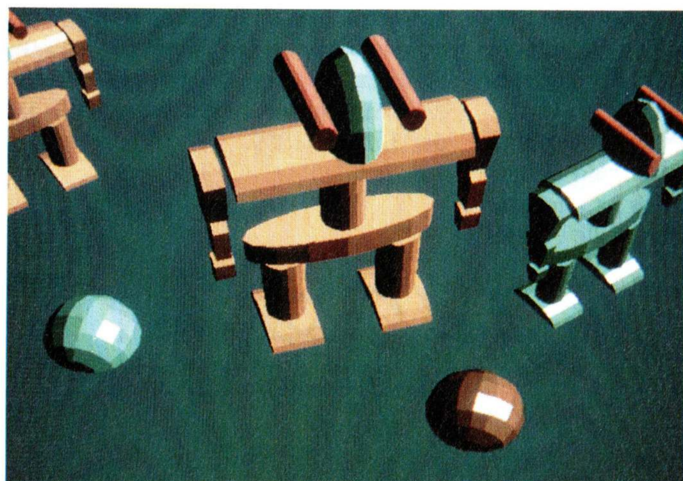
Notepad

Amigawelt

Hard- und Software



Public Domain



High-Tech-Klasse beweist Amiga, wenn er mit der 3D-Software Caligari arbeiten darf. Seite 34.



Feine Striche mit Airbrush heben Konturen hervor und ahmen natürliche Strukturen nach. Seite 74.



Amiga Kicks

Know How

Listing

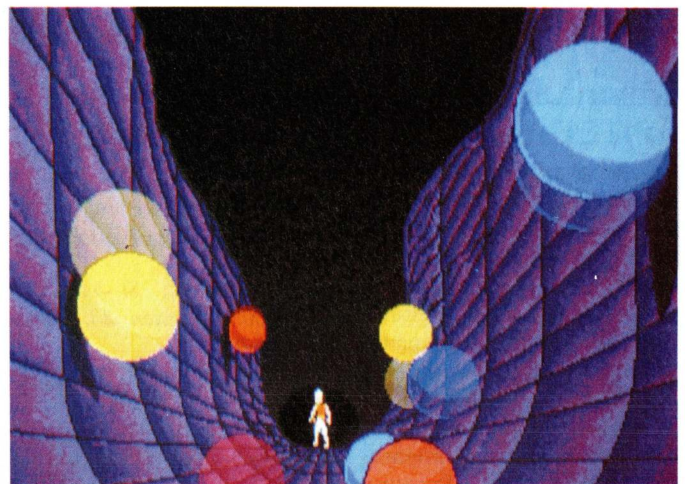
References

Spiele

Rubriken



Viren werden zum Dauerthema für geplagte User. Neues aus der Seuchenforschung auf Seite 90.



Schön bunt, aber äußerst gefährlich: Todeskugeln bedrohen den Helden aus „Dragon's Lair". Seite 145.

Disketten von Roboterhand

Alle fünf Sekunden sieben fix und fertige Disketten. Das schafft Europas modernste Fertigungsstraße. Die Hightech-Anlage des Konzerns Rhone Poulenc steht im romantischen Städtchen Albi/Südfrankreich. Maschinen machen Disketten und der Mensch schaut zu. Zumindest mag es dem nicht eingeweihten Betrachter der vollautomatischen Diskettenproduktion so vorkommen.
Bislang erforderte der Weg vom Polyesterband zur gebrauchsfertigen Diskette viel Raum und viel Handarbeit. Mit einer neuartigen, computergesteuerten Fertigungsstraße wird räumlicher und personeller Aufwand auf ein Minimum reduziert und die Produktivität erhöht.
In 25 Arbeitsgängen werden Disketten gestanzt, ummantelt, geprüft, sortiert, verpackt und verschweißt, ohne daß die Hand eines Menschen eine Diskette berührt. In einer solchen Anlage — genannt HIPAL — wird Rohmaterial — vielfach veresterte Alkohole und Säuren — aus dem RPS-Chemiewerk in Nangi bei Paris verarbeitet. Das filmartige Polyesterband ist mit einer vorübergehend flüssigen, von magnetisierbaren Teilchen durchsetzten Masse beschichtet. Das daraus gepreßte Endlosband wird im Werk Albi in die erste, durch einen Glaskasten staubgeschützte, Maschine geschickt, die daraus runde „Cookies" (= Plätzchen) stanzt. Cookies stellen bereits die eigentlichen Disketten dar.



Die vollautomatische Disketten-Fertigungsstraße steht in einer fast staubfreien Halle. Jede Produktionseinheit ist von einem schützenden Glaskasten umschlossen. Das Personal trägt Spezialschutzkleidung.

Alle weiteren Arbeitsschritte dienen dazu, das Cookie in eine funktionale Mechanik einzusetzen, die Qualität zu prüfen und das Produkt zu verpacken. Das heißt, jede Diskettenscheibe erhält als erstes ein metallenes Mittelstück.

Dann wird weißes, nicht-fusselndes Vlies in achteckiger Form mit einem Loch in der Mitte ausgestanzt und in die Plastikhälften vorgefertigter Diskettenmäntel geklebt. Die Hälften werden zusammengefügt, und in jeden Mantel wird eine Diskette eingeschoben. Das Schreibschutzfenster wird aufgesetzt und mit einer kleinen Feder zum Öffnen und Schließen versehen.

Kleine automatische Schieber überprüfen sofort deren korrekte Funktion. Was nicht einwandfrei arbeitet, wird automatisch aussortiert. Jede Diskette wird daraufhin nach 16 Parametern automatisch auf ihre Zuverlässigkeit untersucht. Dazu werden die Datenträger formatiert und mit einem speziellen Programm bespielt. Automatisch wird geprüft, ob alle Programmfunktionen fehlerlos arbeiten, ob Schreib- oder Lesefehler auftauchen.

Alle Disketten, die den Test bestanden haben, werden gelöscht und die Formatierungsdaten entfernt. Erst bei diesem Prüfvorgang wird nach Qualitäten sortiert. Double-Density- und Single-Density-Disketten werden getrennt produziert. Alle Disketten werden aber double-sided hergestellt; lassen sich jedoch nicht beide Seiten fehlerlos beschreiben und auslesen, sondern nur eine, wandert die Diskette auf den Stapel der single-sided-Disketten.

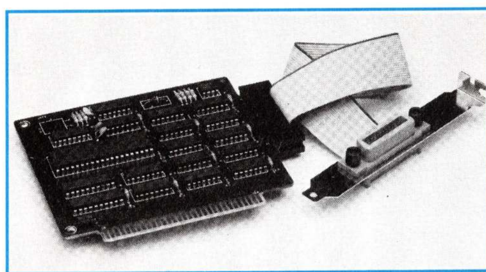
Pro Jahr durchlaufen annähernd 15 Millionen 3½-Zoll-Disketten die 25 Stationen der HIPAL-Anlage. Für die Zukunft plant RPS eine weitere, identische Fertigungsstraße, die ihnen 37 Millionen Francs wert ist. Zur Zeit gibt es lediglich eine weitere Fertigungsstraße dieser Art, und die steht in Japan.

(ub)

Anschluß an IEEE-Welt

Für den Amiga 2000 gibt es jetzt ein IEC-BUS-System mit der Normschnittstelle IEEE 488. Bis zu 14 Meß- und Steuergeräte können gleichzeitig betrieben werden. Alle handelsüblichen IEC-Geräte, dazu gehören die Meßgeräte fast aller namhaften Hersteller wie auch die Peripheriegeräte der alten CBM-8000-Computer können angeschlossen werden. Buskontrolle über Einzelbefehle erlaubt die Anpassung nicht normgerechter Geräte. Die Leistungsdaten sind: vollständige Talker/Listen-Funktion inclusive Remote Control, systemkonforme Software mit Binddriver und autokonfigurierend.

Info: Editronik, St.-Veit-Str. 70, 8000 München 80, Tel.: 0 89/40 40 93



Neues BUS-System macht den Anschluß von Meßgeräten und alten PET-Laufwerken möglich.

Serieller Anschluß vervielfacht

Mit dem Board A2232 sieben RS232-Schnittstellen realisieren. Jede ist dabei individuell auf bis zu 19.2K Baudrate einstellbar. MIDI-DIN-Anschluß befindet sich bereits auf der Karte. Das KSystem ist auch UNIX-fähig.

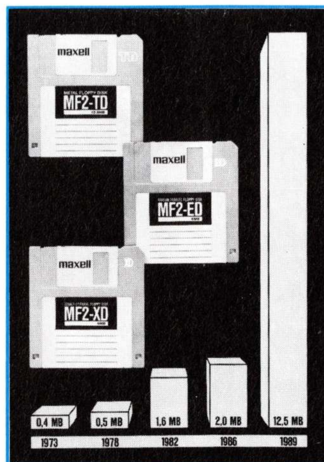
Info: Commodore

Sound- und Grafikhilfe für C-Programmierer

Von der Firma Infotech erhält der C-Programmierer umfangreiche Hilfestellung beim Programmieren und Bearbeiten von Sounds in eigenen Programmen. Umfangreiche Routinen, ein Image-Compiler und Object-Dumps finden sich auf der Diskette. Eine sehr ausführliche Dokumentation erleichtert weiterhin die Arbeit des Bild- und Ton-Bastlers.

Info: Infotech, Thurnerstraße 16, A-4910 Ried, Preis:

Disketten-Zukunft mit höchster Speicherkapazität



Die extrem fortschreitende Entwicklung bei Hard- und Software macht höhere Speicherleistungen bei Disketten erforderlich. Nur „High-capacity"-Disketten sind in der Lage, diese hohen Speicherkapazitäten aufzunehmen. Herkömmliche Beschichtungen und Magnetpartikel sind nicht geeignet, solche hohen Kapazitäten zu speichern. Diesen Anforderungen folgend, stellt Maxell zwei Diskettentypen in 3½-Zoll-Größe mit Speicherkapazitäten von vier und zwölf Megabyte vor. Als Beschichtung werden Barium Ferrite, Cobalt Epitaxial sowie Metal-Magnetpartikel verwendet.

Erst kürzlich wurden von japanischen Laufwerksherstellern entsprechende Diskettenlaufwerke hierzu vorgestellt. So werden die neuen Disketten zum Beispiel in den 12-MB-Floppy-Disk-Laufwerken von NEC und Y.E.Data und in den 4-MB-Drives von Toshiba, Teac und Mitsubishi verwendet. Es ist sicher nur eine Frage der Zeit, wann diese Laufwerkstechnologie auf andere Laufwerkshersteller übergreift und damit der Einsatz von „High-capacity"-Laufwerken zum Alltag gehört. NEC in Japan benutzt bereits ein 12-MB-Laufwerk für ein eigenes Textverarbeitungssystem.

Die Disketten und Laufwerke sind nach unten kompatibel. Das heißt, die Disketten können auch von konventionellen Drives benutzt werden. Diese Laufwerke können auch bislang verwendete Disketten lesen.

„Durch den enormen Anstieg der zu verarbeitenden Daten und die ständige Verbesserung und Ausweitung von Software, bestand eine Notwendigkeit, Disketten mit noch höheren Speicherleistungen zu entwickeln und zu produzieren", kommentiert Karl-Heinz Ewald (Marketing- und Verkaufsleiter Speichermedien bei der Maxell Europe GmbH) die Vorstellung der „High-capacity"-Disketten. „Unsere Forschung und Entwicklung hat diesen Bedarf schon frühzeitig erkannt, so daß wir bereits heute ein serienreifes Produkt präsentieren können", führt er weiter aus.

Im einzelnen weisen die neuen 3½-Zoll-Disketten folgende Spezifikationen auf: MF2-TD — Metall-Beschichtung, Speicherkapazität 12,5 MB unformatiert, 36 700 BPI Aufzeichnungsdichte, 406 TPI Spurdichte, Koerzitivität 1500 Oe; MF2-ED — Barium-Ferrite-Beschichtung, Speicherkapazität vier MB unformatiert, 34 900 BPI Aufzeichnungsdichte, 135 TPI Spurdichte, Koerzivität 1000 Oe; MF2-XD — Cobalt-Epitaxial-Oxyd-Beschichtung, Speicherkapazität vier MB unformatiert, 34 900 BPI Aufzeichnungsdichte, 135 TPI Spurdichte, Koerzitivität 1000 Oe.

Die Kürzel in der Typenbezeichnung geben Auskunft über die Aufzeichnungsdichte. So bedeutet TD „Triple Density" und von „Extra Density" sind ED und XD abgeleitet.

Man ist sich bei Maxell sicher, daß durch die erhöhten Speicherkapazitäten bei Disketten die Hardware- und Softwarehersteller schon bald mit Innovationen aufwarten werden.

Schülerferien im Camp — der Computer ist mit von der Partie

Die Idee zu den Feriencamps kommt aus den USA. In den Camps werden im Rahmen eines ausgefeilten Freizeit-Programms die unterschiedlichsten Lehrgänge in den Bereichen Computer, Sprachen und Sport angeboten. Der Reiz dieser Konzeption liegt in der Vielzahl der Anregungen.

Vor nunmehr sechs Jahren hat *Compu-Camp* diesen zukunftsweisenden Gedanken aufgenommen. Inzwischen haben in den Computercamps Tausende von Jugendlichen die Faszination der neuen Technologie kennengelernt.

Die Kurse sind in Könnensstufen eingeteilt und bauen weitgehend aufeinander auf. Bei der richtigen Kurswahl ist die Aneignung selbst schwieriger Themen spielerisch einfach. Neue Wissensgebiete in gelöster Urlaubs-Stimmung kennenzulernen ist ein spannendes Ferien-Abenteuer.

Je nach Standort und individueller Kurswahl liegen die Kosten für einen Campaufenthalt für eine Woche ab 675 Mark. Im Preis sind Unterbringung, Verpflegung, Kosten für einen Kurs und Freizeitbetreuung eingeschlossen. Interessenten erhalten den Gratis-Prospekt über die Ferien-Camps bei:

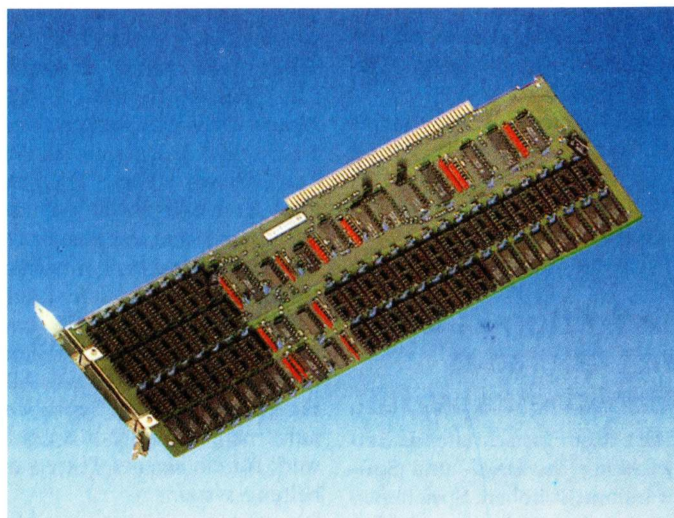
Info: CAMPS, c/o Compu-Camp GmbH, Wedeler Landstraße 93, 2000 Hamburg 56, Tel. (0 40) 81 10 81.





Medl bietet einen Mausersatz für alle unaufgeräumten Arbeitstische an. Info: Medl.

Mit Megabit-Chips zum Speicherausbau



Neue Chips erlauben zügigen Speicherausbau.

8-MB-RAM-Erweiterung erlaubt das A2058/2-Board für die 2000er. Bereits bestückt mit zwei MB, kann die Platine mit weiteren Megabit-Chips erweitert werden. Nach Einstecken in den Slot konfiguriert sich der neue Speicher beim Systemstart selbst.

Info: Commodore

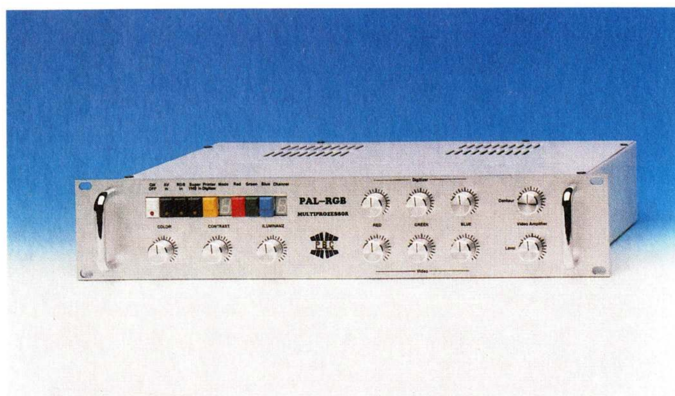


Was am PC schon nützlich war, gibt es jetzt auch für Amiga: Mini-Scanner für Schwarz/Weiß oder 16 Graustufen. Passend dazu gibt es Erkennungs-Software für ASCII-Texte. Info: Reis-Ware, 5584 Bullay.

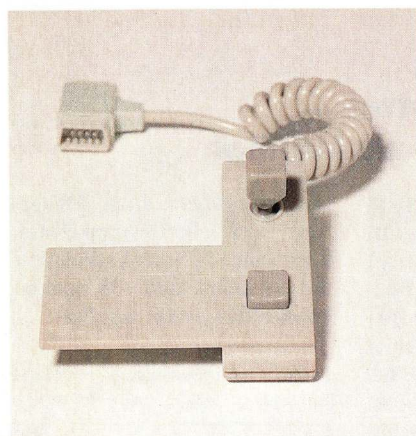
Multifunktionsgerät für den professionellen Videoeinsatz

Mit dem PAL-RGB-Multiprozessor erreichen Amiga-Signale Broadcasting-Qualität. Somit ist der absolut professionelle Einsatz dieses Grafikcomputers bei den Fernsehanstalten kein Problem mehr. Das Gerät verfügt über RGB-Splitting und macht Farbscheiben beim Digitalisieren überflüssig. Mit dem Color-Prozessor lassen sich nachträglich die eingelesenen Farbwerte verändern. Ein Umwandler produziert aus RGB-Signalen FBAS- oder auch Super-VHS-Impulse oder auch umgekehrt. Ferner dient der Multiprozessor als ausgezeichnetes Videoüberspielgerät mit vielfältigen Einstellmöglichkeiten. Zur fernsehgerechten Aufbereitung von Computersignalen ist der Multiprozessor ein unersetzliches Tool.

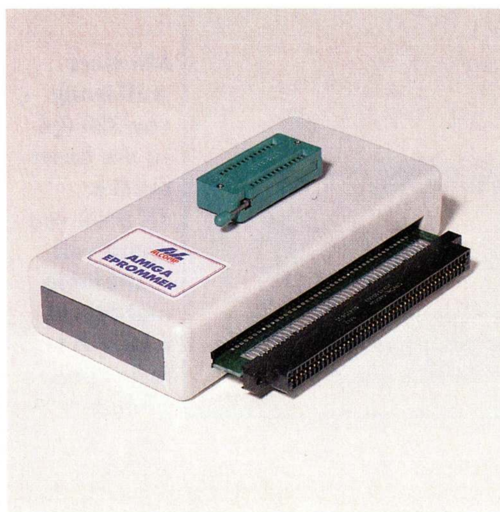
Info: PBC Peter Biet, Georg-Fischer-Str. 5, 6415 Petersberg 2, Preis: 1598 Mark



Zwischen Amiga und Fernsehstation übermittelt dieses Gerät das optimale Signal.

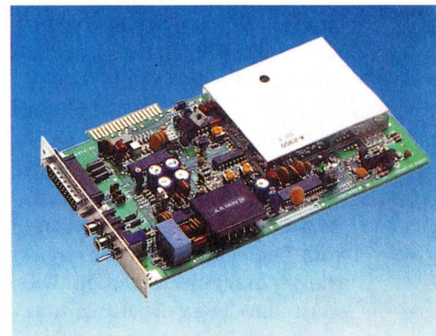


Nicht sehr belastbar, dafür immer parat, ist der Icon-Trotter von Rushware (Kaarst).



Neue Kickstart-Versionen läuten die Zeit der Amiga-Epromer ein. Zum Beispiel der V1.1 von Laufermann, Bedburg.

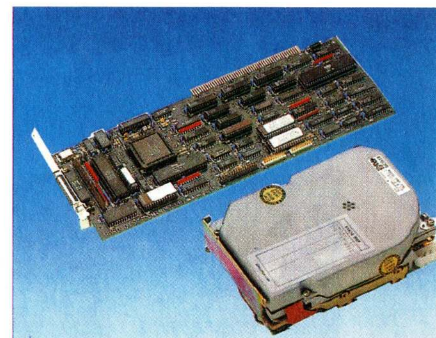
Videokontakt via Steckkarte



Das Genlockboard A2300 ist eine Einfachlösung für Amiga-2000-Besitzer. Im Videoslot steckend, unterstützt es die Bildmischung eines externen Videosignals mit dem Computerbild. Die Karte für den semiprofessionellen Bereich ist in PAL und NTSC erhältlich.

Info: Commodore

A2090A heilt Kinderkrankheit der Harddisk



Endlich autobootfähig sind jetzt die Harddisks im Amiga 2000 mit dem Controller A2090A. Bis zu zwei ST-506-Laufwerke oder bis zu sieben SCSI-Festplatten lassen sich anschließen. Mit Kickstart 1.3 im ROM entfällt das lästige Booten von Diskette.

Info: Commodore



Mit einer Auflösung von 300 dpi ist der Laserdrucker JX 9300 von Sharp voll grafikfähig und druckt 6 DIN-A4-Seiten pro Minute

Monitor A2024 für optimalen Desktop-Einsatz

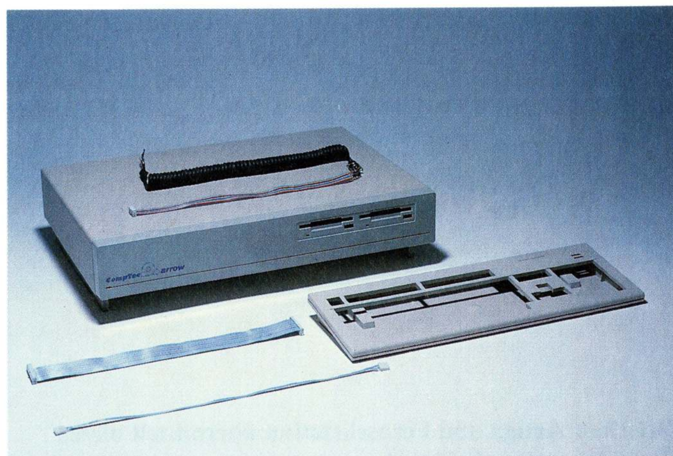
Der monochrome Hedley-Monitor, von Commodore A2024 genannt, wird Desktop Publishing mit Amiga erst zu einer ernsthaften Applikation machen. Darstellungen bis zu einer Auflösung von 1008 × 1024 Bildpunkten sind im non-interlaced-Modus möglich. Die notwendige Grafikhardware ist im Monitor selbst untergebracht. Mit der dazugehörigen Software arbeitet der Monitor mit Amiga 500, 1000 und 2000, sofern sie über mindestens 1 MB Speicher verfügen. Neben Software, die die hohe Auflösung unterstützt, wird auch normale Software verarbeitet.

Info: Commodore

Splitting Images für Digitalisierer

Der neue RGB-Splitter von Creative Video trägt die Typenbezeichnung RGB2 und wurde speziell für den DIGI VIEW entwickelt. Der RGB-Splitter zerlegt das Videosignal einer Farbkamera oder eines Videorekorders in seine drei Grundfarben, welche anschließend vom DIG VIEW in drei Durchgängen digitalisiert werden. Die sonst zur Digitalisierung notwendigen Farbscheiben entfallen. Der RGB-Splitter wird mit je einem Kabel an den Parallel-Port und an den Joystick-Port angeschlossen. Der DIGI VIEW selbst wird einfach auf den RGB-Splitter aufgesteckt. Der sonst für den Amiga 500 und 2000 notwendige Gender Changer ist bereits im RGB-Splitter eingebaut und sorgt unter anderem für die Spannungsversorgung des DIGI VIEW. Über den Joystick-Port erfolgt eine vollautomatische Steuerung des RGB-Splitters. Um die Digitalisierung zu starten, muß im DIGI-VIEW-Menue nur AUTO (ab SW 3.0) angeklickt werden. Etwa 50 Sekunden später erscheint das fertig digitalisierte Bild am Monitor. Damit ist unser RGB2 der erste vollautomatische RGB-Splitter für den DIGI-VIEW. Die Qualität der digitalisierten Bilder übertrifft alles, was bislang für möglich gehalten wurde.

Info: Creative Video, Wolfenäckerstr. 41, 8551 Hemhofen, Telefon (0 91 95) 27 28, Preis: 498 Mark



Neue Karosserien für 500er lösen Platz- und Imageprobleme von Amiga-Home-Usern (Computec, 4192 Kalkar).

Mehr als flackerfrei — Großbildschirm als Monitor

Von der Firma Logotec kommt eine Commodore-Auftragsarbeit. Der Großmonitor Viking 1 stellt auf einem 20-Zoll-Bildschirm in vier Graustufen flikkerfreie Grafik bis zu einer Auflösung von 1008 × 800 dar. Bei einer Bildwechselfrequenz von 63 Hz ist das Bild nicht nur flicker- sondern auch flackerfrei. Zwei volle Desktop-Seiten oder eine Tabellenkalkulation mit 68 Zeilen und 138 Spalten können dargestellt werden.

Info: LOGOTEC, Humboldtstr., 2056 Glinde, Tel.: 0 40/7 27 70 50



Creativ-Video rüstet Digi-View mit R6B-Splitter aus.

Noch eine Textverarbeitung für den Amiga

Scriptum will hohen Ansprüchen genügen. Ob schnell ein Brief oder in Ruhe ein längerer Text geschrieben werden soll: Scriptum verspricht leichte Bedienbarkeit. Die Länge des Textes hängt nur von dem freien Speicher Ihres Amiga ab. Scriptum enthält alle Funktionen zur professionellen Gestaltung eines Textes. Man kann Text einfügen oder überschreiben, Tabellen mit Tabulatoren erzeugen, Schrift formatieren (wie fett, kursiv, hoch- oder tiefgestellt), Kopf- und Fußzeilen definieren, und vieles mehr. Hardware-Anforderungen: Amiga 500, 1000 oder 2000 mit mindestens 512 KByte Speicherkapazität.

Info: M & T, Haar

Video-Schnappschüsse mit SNAPSHOT

Das neuartige Video-Digitizer-Hard- und -Softwarepaket SNAPSHOT mit integriertem RGB-Splitter ermöglicht problemloses Digitalisieren mit dem Amiga.
Beliebige Schwarzweiß- oder Farbvorlagen lassen sich mit Hilfe Ihrer Videokamera konturenscharf und mit brillanten Farben darstellen.
SNAPSHOTs hervorstechende Vorteile: digitalisiert ohne lästigen Farbfilterwechsel die Bilder von Farbkameras automatisch und mischt sie in Sekunden zu einem HAM-Farbbild mit 4096 Farben; Bildhelligkeit, Bildkontrast, Farbton und Farbsättigung können über weite Bereiche eingestellt und wie die Bildschärfe sofort auf dem Amiga-Monitor kontrolliert werden; menügesteuerte Auswahl der sechs Bildauflösungen; hundertprozentiges IFF-kompatibles Bildspeicherformat; arbeitet mit jeder Schwarzweiß- oder Farbkamera zusammen, das heißt auch mit den beliebten Camrekordern; sinnvoller Einsatz von fünf Auflösungen schon ab 512 KByte Speicher möglich (HAM 320×512 ab 1 MByte).

Info: Binnewies, Hannover, Tel.: 05 11/43 10 06; Preis: 498 Mark.



SNAPSHOT-Schnappschüsse machen auch auf dem Bildschirm einen guten Eindruck

Amiga-Entwickler tagten in Frankfurt

Nach Commodores Absage im Herbst '88 trafen sich Mitte Januar in Frankfurt die Hard- und Softwarekonstrukteure zur zweiten europäischen Developers Conference. Mit amerikanischen Kollegen diskutierten sie die aktuellen Neuheiten und entwarfen Zukunftspläne.



Amiga World-Chefredakteur Guy Wright erinnerte die Entwickler daran, daß die Interessen der Enduser Berücksichtigung finden müssen.



Andy Finkel, verantwortlich für die Systemsoftware, nahm viele Anregungen europäischer Entwickler mit nach West Chester

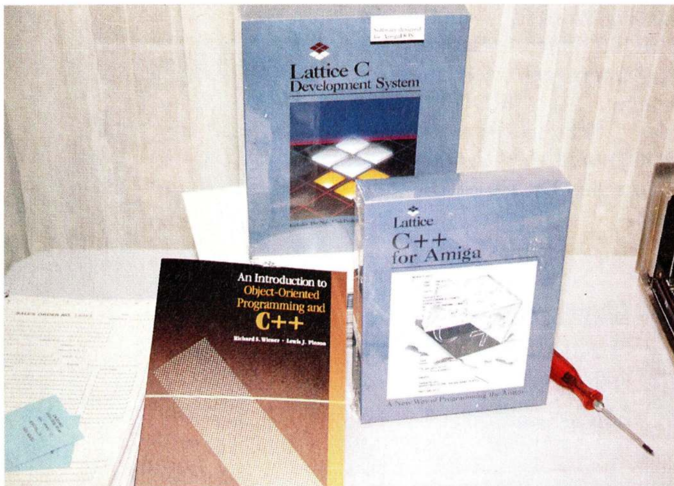
Entwicklerkonferenzen dienen dem Erfahrungsaustausch von Hard- und Softwareingenieuren untereinander und der Interessenabstimmung mit den Herstellerfirmen. Für Amiga war eine derartige Kommunikationsbühne lange überfällig. Daher bemühten sich zum zweiten europäischen Entwicklertreffen vom 16. bis 18. Januar neben den Braunschweiger Computerköchen auch die wichtigsten amerikanischen Vertreter von Commodores Entwicklerküche: Andy Finkel, verantwortlich für das Systemsoftwaremanagement und die Weiterentwicklung des AmigaDOS, Jim Mackraz, Programmierung der Grafikchips und Grafiksoftware, betriebssystemkonformes Verhalten, Bob „Kodiak" Burns, Inputhandler und Werkzeuge, Wolf Dieter Schmidt, Transputer und Johann George, der maßgeblich an der Unixportierung mitwirkte. Vom Marketing waren Gale Wellington, Jeff Porter und Rick Glover anwesend, ferner John Toebes von Lattice, der den Lattice C Compiler entwickelt, und Tom Rokicki, der TeX auf den Amiga portierte.
Um die Gerüchteküche über utopische neue Produkte für den Amiga auszuräumen, traten Jeff Porter und Andy Finkel gleich zu Anfang der Konferenz auf das Podium und schafften Klarheit, welche Neuheiten dieses Jahres den Markt erreichen werden.
Um vor allem für Desktop Publishing und Computer Aided Design einen hochauflösenden und papierweißen Monitor anzubieten, entwickelte Commodore den Hedley Monitor A2024, der an jeden Amiga mit mindestens 512 KByte Chip-RAM und 512 KByte Fast-RAM anschließbar ist. Der Monitor bietet vier Graustufen weiß, hell- und dunkelgrau und schwarz. Die Auflösung beträgt 1008 mal 1024 Punkte. Von dem Monitor wird es zwei Versionen geben; die Commodore-Version ist ein 15-Zoll-Monitor, in dem die Grafikhardware eingebaut ist. Eine weitere 20-Zoll-Version wird von einem deutschen Dritthersteller für den 2000er angeboten, bei der die Grafikhardware auf einer Steckkarte aufgebaut ist. Um den Monitor anzusteuern, benötigt man Software, die in die Betriebssystemversion 1.4 integriert werden wird. Mit der Auslieferung für den Privatkunden ist erst mit Erscheinen der Betriebssystemversion 1.4 im nächsten Jahr zu rechnen.
Jetzt schon ausgeliefert wird das Genlock Interface A2300 für den Amiga 2000, das für wenig Geld brauchbare Leistung anbietet.
Brisant ist der Professional Video Adapter A2350, der die Fähigkeiten eines Framegrabber und eines Genlockinterface auf zwei Karten bietet. Die Hardware besitzt eigenes RAM, in das ein Fernsehbild ohne Qualitätsverlust eingelesen werden kann. Software verwandelt hinterher die Grâfik in beliebige Amiga-Formate, um die Bilder mit Amigasoftware nachbearbeiten zu können — ein Produkt für Videoprofis.
In Verbindung mit dem neuen Unix-Betriebssystem (siehe dieses Heft) wurde die Turbo Karte A2620 mit 32-Bit-Architektur entwickelt. Sie bietet einen 68020 Prozessor mit FPU und PMMU und zwei bis vier MByte RAM und wird mit einer Taktfrequenz von 14,3 MHz betrieben. Der Mathecoprozessor kann optional durch eine 68882 mit 20 oder 25 MHz ersetzt werden. Der Harddiskkontroller A2090A ist mit Kickstart 1.3 autobootfähig. Besitzer des alten A2090-Controller können mit Hilfe der Zusatzplatine A2090B, die in irgendeinem 100-Pin-Amigaslot ihren Platz findet den A2090-Kontroller bootfähig machen. Die neue A2058-RAM-Erweiterung erlaubt den Speicherausbau bis 8-MByte-Fast-RAM auf einer einzigen Platine mit MBit Chips.

Profi-Erweiterungen für 2000er

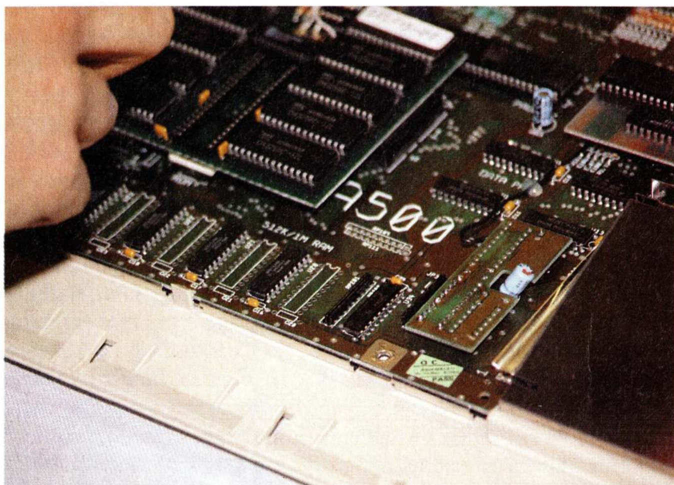
Für die Besitzer eines Amiga 500 gibt es einen kombinierten SCSI-Harddiskcontroller und 2 MByte Speichererweiterung A590, die in einem kompakten Gehäuse die beiden größten Erweiterungswünsche aller Amigabesitzer einfach wahr werden läßt. Als Festplatte wird eine sehr preiswerte und deshalb aber auch langsame 20-MByte-SCSI-Platte mitgeliefert.
Für professionelle Anwendungen wird eine Erweiterungskarte mit sieben seriellen Schnittstellen angeboten. Um den Prozessor des Amigas zu entlasten, ist ein 6502 Einprozessorcomputer auf der Multiseriellplatine vorhanden, der Daten mit bis zu 19,2 KBaud senden und empfangen kann.
Das von allen ungeduldig erwartete Enhanced Chip Set bietet Kompatibilität zu den alten Chips und besitzt einen neuen Grafikmodus, der „Productivity Mode" genannt wird. Er besitzt eine Auflösung von 640



Mit dem 5-RAM von Combitec kann der Amiga 500 erstmals von einer nichtflüchtigen Speichererweiterung Workbench und Software booten.



C++, das Entwicklungspaket für komfortable C-Programmierung, präsentierte sich den Entwicklern in marktreifer Form.



Unter den Erweiterungsplatinen von Flesch + Hörnemann verbarg sich eine aufgeräumte Grundplatine — vorbereitet für die Serienproduktion von 1 MB – 500er.

mal 512 Punkten mit vier aus 64 Farben pro Punkt. Das ECS wird zusammen mit der Systemsoftware 1.4 seine Markteinführung erleben. Bis jetzt sind nur handgeschnitzte Testmuster mit Betaversionen der neuen Systemsoftware in den Händen einiger Entwickler zugänglich.

Verschiedene Hard- und Software-Entwickler zeigten Muster ihrer neuen Produkte. So stellte CIT eine Grafikkarte mit einem Vektorgrafikprozessor und einer maximalen Auflösung von 1280 mal 1024 Punkten mit 16 Farben aus 256 000 pro Punkt vor. Die TU Graz zeigte eine Btx-Software, die es mit einem handelsüblichen Modem ohne weitere Zusatzgeräte auf einem Amiga mit 512 KByte ermöglicht, am Btx-Dienst der Post teilzunehmen. Flesch und Hörnemann präsentierte eine Kickstartumschaltplatine für den 500er. Wahlweise kann das Kickstart aus dem ROM benutzt werden, oder wie schon beim Amiga 1000 in 256 KByte Spezial-RAM von Diskette eingeladen werden.

Die Tagungsbeiträge der Redner boten dem Zuhörer die Möglichkeit, sich schwerpunktmäßig über Marketing, Einführung in die Programmierung oder Profi-Know-how zu informieren. Wichtiger allerdings als die Vorträge waren die Gespräche, die man anschließend mit den Entwicklern über Probleme führen konnte bis hin zu Diskussionen über spezielle Programmierprobleme, die bis in die frühen Morgenstunden im Computerraum ausgetragen wurden.

Commodore schätzte die Wichtigkeit und Kompetenz der europäischen Programmierer richtig ein und konnte viele wertvolle und konstruktive Vorschläge speziell für die nächste Betriebssystemversion entgegennehmen.

Als Ergebnisse für die Version 1.4 kann folgendes festgehalten werden:

— Das Enhanced Chip Set wird voll unterstützt.
— Der hochauflösende Monitor wird voll unterstützt.
— Es werden Funktionen implementiert, die das Umwandeln von Bitmaps verschiedener Auflösungen ineinander bieten, um neue Grafikauflösungen benutzen zu können. Möglicherweise wird es später etwas ähnliches wie Display Postscript geben.
— Intuition und die Layers Library werden grundlegend überarbeitet, um die vielen verschiedenen Grafikauflösungen leicht und kompatibel programmierbar zu machen, und um eine höhere Arbeitsgeschwindigkeit zu erreichen.
— Preferences wird total überarbeitet.
— Die Wortbench wird eine neue Oberfläche erhalten und wesentlich mehr Möglichkeiten zur Manipulation von Files ohne Icon bieten.
— Serial- und Parallel-Device werden an die Handhabung mehr als einer Schnittstelle jeder Art in einem Rechner angepaßt.
— Das Amiga DOS benutzt Fast File System standardmäßig auch mit Disketten, erkennt aber weiterhin Disketten des alten Formats ohne Probleme.

BCPL soll bis zur Version 1.5 völlig aus dem DOS verschwinden. Ein Standard-Filerequester wird dem Programmierer zur Verfügung gestellt. Die für den Programmierer zugängliche DOS-Library wird erweitert. Die Shell und die CLI-Commandos werden überarbeitet und verbessert, möglicherweise wird ARexx ein fester Bestandteil des Betriebssystems.

— Color Fonts werden durch das Betriebssystem unterstützt.
— Schnellere Text-Routinen werden in das ROM integriert.
— Noch mehr Fließkommastandards werden unterstützt.

Die DevCon war für alle Teilnehmer ein voller Erfolg, und es ist zu hoffen, daß es weitere regelmäßige Treffen dieser Art geben wird. Lediglich die Interessen der Enduser sollten stärker Berücksichtigung finden.

(Wilfried Häring)

MIDI-Märchen aus Beatles-Songs

Bei Norwegian Wood handelt es sich um eine LP-Produktion, die ein Symphonie-Orchester, Rockmusiker, Computersound und -grafik zu einer homogenen Kombination vereint. Das Projekt wurde mit einem Amiga 2000 verwirklicht.

Arrangeur R.A.M. Pietsch stellte klare Anforderungen an seinen Rechner: er soll hauptsächlich im Tonstudio als MIDI-Aufnahmesystem und als Klangbearbeitungsstation für neue synthetische Klänge eingesetzt werden.
Für den Einsatz als Recordingsystem war an die zu verwendende Musiksoftware als Kriterium die Gewährleistung der Synchronisation des Computers via SMPTE mit mehreren 24-Spur-Bandmaschinen zu sehen und weiterhin die Bereitstellung von mindestens 30 Spuren im MIDI-Recordingprogramm.
Weiterhin war für die Produktionsmethode wichtig, die Multitaskingmöglichkeit des MIDI-Recordingsystems gleichzeitig mit den Synthesizereditorprogrammen zu nutzen. Da Musiker ja oft während der Studiosessions mit neuen Sounds und Soundkombinationen und den verschiedensten Synthesizern experimentieren, war diese Anforderung der Multitaskingfähigkeit an das System wesentlich.
Deshalb fanden die Künstler den Amiga mit 3 MB Hauptspeichererweiterung als Ideallösung für die Verwirklichung des Projektes. Für die Lösung aller anderen Anforderungen stellten die Münchner Musik-Computerfachmänner von Multi-Media die Programme und Hardware kostenlos zur Verfügung. Die Entscheidung fiel auf das MIDI-Recordingsystem der amerikanischen Firma Dr. T's (Kurzform: KCS.TM V1.6-Keyboard Controlled Sequencer) das mit 48 frei verfügbaren MIDI-Spuren, 128 Patterns, 16 Songs gleichzeitig, der Multitaskingfähigkeit und der SMPTE.Synchronisation die Anforderungen leicht meistern konnte. Als MIDI-Interface, also Schnittstelle zwischen Amiga und den Synthesizern, diente das Dr.T's MIDI-Interface.


Klaus Voormann, R.A.M. Pietsch und Frank von den Bottlenberg im Soundstudio

Blitzschnelle Soundbearbeitung mit grafischem Musikeditor

Eine Synchronisationsbox wurde für die SMPTE-Synchronisation von den Bandmaschinen zum Amiga benötigt. Das SM-9 hat sich in der Produktion hervorragend bewährt und überzeugte durch einwandfreies Funktionieren, das heißt korrektes und schnelles Anlocken an den SMPTE-Code, der ja von den Bandmaschinen kommt.
Im Studiobetrieb kommt es immer auf Geschwindigkeit und Flexibilität an, welche den Amiga deshalb so interessant in der Musikanwendung macht. Wenn sich der Produzent im Studio noch nicht sicher ist, mit welchen Klängen oder Geräuschen er die einzelnen Sequenzen unterlegen will, kann er mit Hilfe der diversen Synthesizereditorprogrammen die verschiedensten Sounds in Windeseile austauschen oder nur geringfügig verändern. Die Künstler benutzten wieder die bewährten Dr. T's-Programme im Multitaskingbetrieb mit dem Sequenzer KCS V1.6. Für unser Yamaha-TX-816-Rack und DX-7 verwendeten wir Dr.T's DX-Heaven, das Editorprogramm für alle Yamaha-6-Operatorensynthesizer. Mit Hilfe dieses Editorprogrammes konnte man innerhalb von Sekundenbruch-


Hans Rüdiger Lutz designte das Cover

Sound und Cover vom Computer

R.A.M. Pietsch wählte für seine Produktion drei Beatles-Songs aus, die thematisch durch ein Märchen verknüpft werden. Schauplatz des Märchens ist ein seelenloser, totalitärer Staat. Erzählt wird die Geschichte eines Landvermessers, der den Abbau der wenigen verbliebenen Natur meßtechnisch vorzubereiten hat. Während seiner Arbeit dringt er in fremde Gefilde vor; dort lernt er ein Mädchen aus einer anderen, gefühlvollen Welt kennen und lieben. Mit ihrer Hilfe gelingt es ihm, die seelenlose Gesellschaft, der er ent-

teilen Sounds austauschen, oder auch mit Hilfe eines Zufallsgenerators neue Sounds generieren. Für den Roland D-50 und den D-550-Soundmodule konnten sie mit dem Dr. T's D50-Editorprogramm sehr schön – auch grafisch mit Hilfe der Maus – sofort Sounds bearbeiten und für alle Bedürfnisse anpassen. Die Anwendung des KCS 1,6 als MIDI-Recordingsystem hat sich bewährt. Der Amiga ermöglichte es, mit dem gesamten MIDI-Equipment zu jedem Zeitpunkt umzugehen, egal ob er gerade zu den Bandmaschinen synchronisiert wurde, oder mit MIDI-Exclusiv-Daten gearbeitet wurde. Das Recordingsystem wurde mit einer Auflösung von 960 Ticks pro Takt definiert. Diese hervorragende Auflösung garantierte eine wirklich realitätsgetreue Wiedergabe aller eingespielten Spuren.

So war es zum Beispiel dann auch möglich, mit Hilfe der 48 Spuren wirklich mehrere Variationen der einzelnen Parts zu speichern und dann die besten Passagen zusammenzusetzen.

stammt, zu revolutionieren. Die Entstehung dieser Welt wird von Musik untermalt – von Musik, die zuerst von nicht-elektronischen Instrumenten gespielt, später aber durch Computerklänge ergänzt wird.

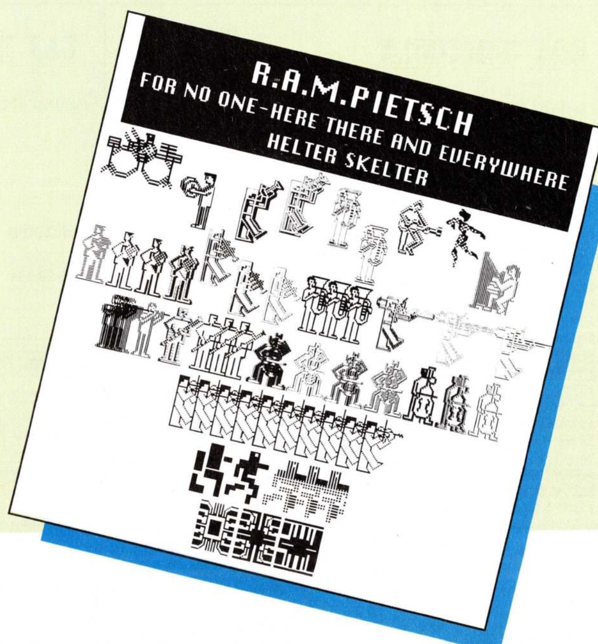
Die von R.A.M. Pietsch bearbeiteten Beatles-Klassiker sind jetzt auf einer Maxi-Single mit dem Titel „For no one – here, there and everywhere – helter skelter" erschienen. Doch der Computer hat nicht nur Einzug in die Musik gehalten, sondern auch – auf grafischer Ebene – in das Cover-Design. Das LP-Cover wurde von Klaus Voormann, Beatles-Intimus, entworfen. Klaus Voormann erhielt bereits einen Grammy für die Gestaltung des Covers der Beatles-LP „Revolver".

Technische Grundlage für das Design war „DeluxePaint" von Electronic Arts. „DeluxePaint" ist unkompliziert, schnell und professionell; zudem bietet es die Möglichkeit, spielend eine kontrollierte Zufallsgestaltung zu realisieren. So hat sich auch Hans Rüdiger Lutz, Dozent an der Akademie der bildenden Künste, Stuttgart, für dieses Equipment entschieden, um seine „digitalen Musiker", die auf dem Maxi- und Single-Cover in vielen Variationen zu sehen sind, zu produzieren. Zuerst konstruierte er die Umrißlinie der einzelnen Musiker auf einem Raster von 4 × 4 mm; mit Hilfe der Scribbles entstanden schließlich die definitiven Computerbilder. Mit dem „Brush" wurden die Figuren noch etwas verkleinert; sie konnten dann auf der gesamten Fläche versetzt werden. So war es möglich, die eingefangenen Motive zu dehnen, zu quetschen, übereinanderzulegen und zu stanzen, weiterhin Duplexe, Triplexe zu machen und die neuen Rastermutationen zu reihen und zu isolieren.

Das Ziel dabei war, die Figuren möglichst chipdesigned und „digital" aussehen zu lassen. Viele wurden sofort gestanzt und outlined, bis sie Leiterbahnen und Platinen glichen. Entsprechend wichtig war es, die Produkte nach optischen und ästethischen Gesichtspunkten zu kontrollieren. Als letzter Schritt wurden die ausgewählten Figuren lasergedruckt und klassisch als Reinzeichnung montiert.

Das Ergebnis ist auf dem Cover der Maxi-Single „For no one – here, there and everywhere – helter skelter" von R.A.M. Pietsch (spv 50-8811) zu sehen.

(Christian Thalhammer)



Ordnungshilfe für PD

Zehn Inhaltsetiketten, passend zu den Beschreibungen in diesem Heft, sorgen für Ordnung in der PD-Schatztruhe. In der nächsten Ausgabe geht es weiter, auch ältere Fish-Disks werden nachgetragen. Daher: jetzt anfangen zu sammeln!

FishDisk 169

▶ schreiben
▷ schützen

FishDisk 169



FishDisk 169

Matt Dillon's C-Utilities, Vol. 2

DME	Source zu FD 168
SUPLIB	Binär-Code und Source, Docfile auf FD 168
DRES	Binär-Code und Source, Docfile auf FD 168
Shell	Source zu FD 168

FishDisk 163

▶ schreiben
▷ schützen

FishDisk 163



FishDisk 163

Bankn	Scheckbuchsystem (Shareware, Update von FD 120, binär)
FiveInLine	Brettspiel ähnlich wie Go-Moku o. Ristinolla (mit Sources)
MachII	Mausprogramm mit Hotkey, sunmouse-Features, clicktofront, popcli, titlebar und anderes (Update zu FD 130, binär)
MemTrace	Routinen, um belegten Speicher wieder freizugeben
PcPatch	Patches für PCCopy und PCFormat von "Extras 1.2", um Zugriff auf 3,5 Zoll-Disketten unter MS-DOS zu ermöglichen.
ReadmeMaster	Kleine Database, um nach Programmen auf den FishDisks zu suchen.
View	Mausgesteuerter Textanzeiger

FishDisk 166

▶ schreiben
▷ schützen

FishDisk 166



FishDisk 166

AutoGraf	Grafische Benzinkostenabrechnung (mit Source)
Cref	Ein C-Cross-Referencer
MultriCalc	Taschenrechner-Emulation. Angaben bis zu 3000 Stellen hinter dem Komma (nur binär)
Stevie	Clone des UNIX-Editors „vi" (mit Sources)

FishDisk 170

▶ schreiben
▷ schützen

FishDisk 170

FishDisk 170

Aftterm	Kommunikationsprogramm (nur binär, IBM 3278 Emulation)
Dis6502	Dissassembler für C 64-Binärfiles
FastText	Schnelle Textanzeige auf Blitter-Basis (mit Source und Demo)
MRBackup	Fileweises Backup der Festplatte
PtrAnim	Kleines Animationsprogramm (binär, Shareware)
Surf	Produziert gewölbte Oberflächen, auf denen sich IFF-Bilder darstellen lassen (mit Source)
Turbo	Über ein Gadget lassen sich Bitplane, Sprite, Copper und Audio-DMA abschalten, um die Systemgeschwindigkeit zu erhöhen

FishDisk 164

▶ schreiben
▷ schützen

FishDisk 164



FishDisk 164

C-Functions	Vier nützliche C-Routinen für Programmierer (Sources und Demos)
Disksalv	Rettet versehentlich gelöschte Diskettenfiles, sofern sie noch nicht überschrieben sind (Update zu FD 20, zwei Laufwerke erforderlich, binär)
Hed	Leicht zu bedienender Editor, dennoch leistungsfähiger als „Ed" (binär)
Newton	Nach der „Newton-Methode" wird der reale und imaginäre Teil eines Polynomial von 20 Grad und weniger geschätzt (mit Source)
NewZap	Utility, um File-Sektoren zu editieren (Update zu FD 58, binär)
PcView	Ermöglicht IFF-Bilder mit PC-EGA darzustellen (mit Sources)
PolyRoot	Polynomial-Root-Finder in AmigaBasic
PrtDrivers	Druckertreiber für LNO3+ und MT420d
Zoo	Update des File-Packers von FD 136 (binär)

FishDisk 167

▶ schreiben
▷ schützen

FishDisk 167



FishDisk 167

CDecl	Übersetzt englische Anweisungen in C-Ausdrücke (mit Source)
CLIcon	Damit lassen sich CLI-Programme mittels Icon starten
CloseMe	Screen-Hack (mit Sources)
DSM	Demo-Version von „Dynamic Sound Machine
MRPrint	CLI-Routine zum Ausdruck von Text (viele Features incl. Source)
SMUS3.6a	Update des SMUS-Players von FD 58.
Sounddemos	Stereo-Sound in Perfektion

FishDisk 171

▶ schreiben
▷ schützen

FishDisk 171



FishDisk 171

AZComm	Modemsoftware (Modifizierte Version von Comm 1.34, nur binär)
Maze	Zwei Labyrinth-Demos
Sozobon-C	Freeware C-Compiler
Xoper	Hintergrundprogramm zur Kontrolle von Speicher, Interrupts u. a.

FishDisk 165

▶ schreiben
▷ schützen

FishDisk 165



FishDisk 165

Conman	Sehr nützlicher Standard-Console-Handler (Update zu FD 133, Shareware)
CPM	Emuliert einen CPM-Computer mit Z80 Prozessor an einem ADM3A-Terminal
Parsnag	Unterstützt Farbseparationen auf Epson JX-80-Druckern (Assembler und Source).
PlotView	Hilfsprogramme, um Unix-Plot-Files darstellen zu können (mit Source)
RamCopy	Kopierprogramm für Computer mit 1 MB und nur einem Laufwerk
SPUDclock	Sagt in definierten Intervallen die Zeit an (mit Source)

FishDisk 168

▶ schreiben
▷ schützen

FishDisk 168



FishDisk 168

Matt Dillon's C-Utilities, Vol. 1

CONFIG V1.00	Plaziert nützliche Programme auf der Workbench
Clock V1.00	Eine einfache Uhr
DME V1.31	Editor für Programmierer
DMouse V1.10	Tune-up für den Mauszeiger
Backup V2.01	FestplattenBackup
LIBREF	Support, um Matt's Programme zu compilieren
DRES V1	Support-Library für run time
DASM	Small System Assembler
FTOHEX	Konvertiert DASM-Executables in Intel-Hex
FILES	Disketten-Katalog-Programm
Shell V2.10	Erweitertes CLI
FINDIT V1.00	Sucht nach Filenamen (mit Wildcards)
LIBS V1.00	Auflisten von Libraries und Devices
SCAT V1.00	Utility, um Binär-Code darzustellen
ADDCR V1.00	Fügt vor jedes Linefeed ein Return ein
REMCR V1.00	Entfernt Returns
CMP V1.00	Vergleicht zwei Files

FishDisk 172

▶ schreiben
▷ schützen

FishDisk 172



FishDisk 172

DataToObj	Konvertiert reine Datenfiles (Text, Sprites, Images u. a.) in Objekt-Code
Handshake	VT52/VT100/VT102/VT220-Emulation (nur binär, Shareware)
MFix	Schaltet den Marauder-Regenbogen ab — bis zu 25% schnellerer Kopiervorgang (nur binär)
PopInfo	Anzeige zusätzlicher Informationen auf der Workbench
ProCalc	Simuliert den HP-11c-Rechner (binär, Shareware)
Spiff	Flexibles Programm, um zwei Files miteinander vergleichen zu können (mit Sources und Beispielen)

2 MB Flugzeuglärm

Beim Rüstungskonzern MBB ist vieles so geheim, daß man sich kaum an etwas zu erinnern traut. AmigaWelt-Autor Andreas Fuchs hat bei MBB-Programmierern Aufsehen erregt. Hier sind Teile seiner Erinnerung.

Bei großen Flugsimulatoren, wie dem für den Airbus A310 von MBB, sind Fluggeräusche oder der Ton eines Warnsystems unerläßlich. Da die Simulationshauptrechner jedoch keine Sounds produzieren können, muß dies von einem externen Rechner übernommen werden. Ursprünglich war für diese Aufgabe ein 128 'Diesel' vorgesehen, der über externe Soundkarten die Sounds ausgeben sollte. Dies schien jedoch nur sehr schwer realiserbar, und so wurde ein anderer, besserer Rechner gesucht. In der zuständigen Abteilung bei MBB war einiges über die gute Sound-Fähigkeit des Amiga bekannt, und so begann man sich mehr und mehr für diesen Computer zu interessieren. Allerdings weist der Amiga zwei Defizite auf: nur vier Tonkanäle und max. nur 430 KB RAM, die von den Tonkanälen ansprechbar sind. Der Amiga konnte diesen Part also unmöglich alleine übernehmen. Deshalb wurde entschieden, eine PC-Karte einzubauen.

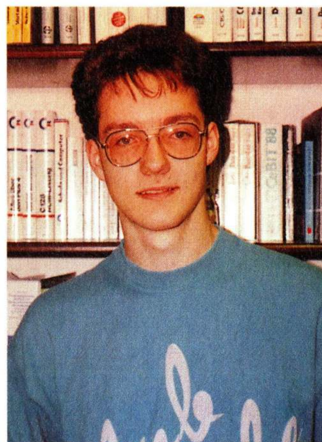
Diese PC-Karte sollte wiederum bis zu 16 Samplerkarten ansteuern können. Die Hardware schien gefunden. Nun überlegte man sich, wie Amiga und PC ihre Informationen bekommen sollten. Es war geplant, daß der Simulationshauptrechner, kurz Host genannt, über eine RS232-Schnittstelle die Steuerdaten für PC und Amiga an den PC schickt. Der PC übergibt dann eine Kopie dieser Liste mit Steuerdaten an den Amiga und spielt selbst schon die Sounds, die nicht vom Amiga gespielt werden. Der Amiga sucht sich nun die entsprechenden Daten aus der Liste heraus und spielt den dazugehörigen Sound.

Vorerst war dies für MBB noch Zukunftsmusik, da man noch nicht wußte, wer dieses Programm überhaupt schreiben sollte, denn die Werksprogrammierer kannten sich mit dieser speziellen Eigenschaft des Amiga nicht aus. Zunächst wurde ein Digitizer gekauft, mit dessen Hilfe die Sounds gesampelt wurden. Doch dann mußte langsam daran gedacht werden, eine eigene Abspielroutine zu schreiben. Bei dem Digitizer war sogar ein Source-Listing für eine Abspielroutine dabei, jedoch war dieses Programm viel zu langsam, zu kompliziert und zu umfangreich. Also wurde nach einer schnelleren, kürzeren und einfacheren Lösung gesucht. Da kam den Experten von MBB der Artikel über die ShortSound-Routine in Amiga Welt 4/88 wie gerufen. Nun war endlich nach vielen Inseraten jemand gefunden worden, der sich mit der Soundprogrammierung auf dem Amiga gut auskannte. Also beauftragte MBB mich, eine Soundroutine nach ihren Vorstellungen zu schreiben.

Short Sound überholt kommerzielle Abspielroutine

Damit der PC dem Amiga die Liste der Steuerdaten möglichst einfach übergeben kann, enthält die PC-Karte noch ein „Dual ported RAM". Ein Dual ported RAM ist einfach ein Speicherbereich, auf den sowohl PC als auch Amiga zugreifen können (lesen und schreiben), deshalb wurde er auch treffend Janus (Gott mit zwei Gesichtern) genannt. Der PC schreibt also, nachdem er die Steuerdaten des Hostrechners empfangen hat, eine Kopie dieser Steuerdaten in den Janus. Der Amiga holt sich nun aus dem Janus seine Daten heraus und spielt den entsprechenden Sound. Da es nun aber über 30 Sounds sind (jeder ca. 40 KByte lang), können unmöglich alle Sounds gleichzeitig im Chip-RAM stehen.

Deshalb müssen die Sounds erst geladen werden. Speichermedien wie Floppy oder Festplatte



Andreas Fuchs' Short Sound sorgt für Flugzeuggeräusche.

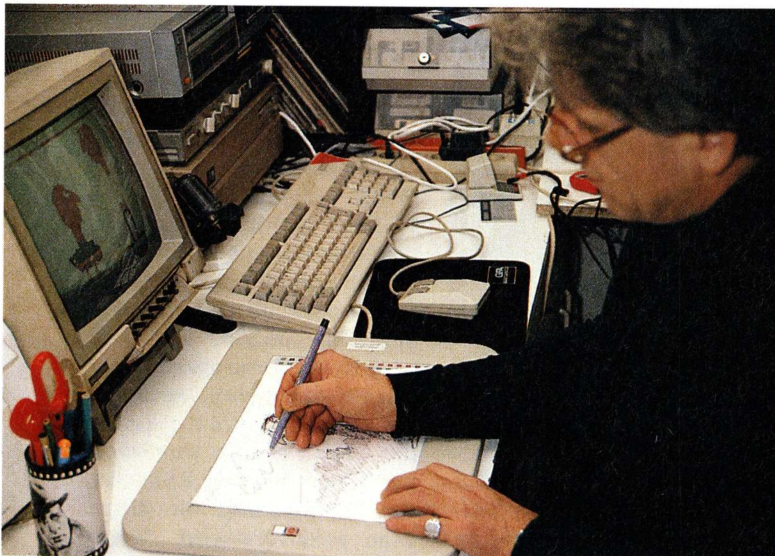
wären jedoch zu langsam, deshalb hat man sich entschieden, die Sounds erst ins Fast-RAM zu kopieren, und dann den benötigten Sound ins Chip-RAM zu laden. Doch auch hier tauchen wieder Probleme auf: So kann zum Beispiel das Kopieren eines Sounds aus dem Fast-RAM ins Chip-RAM nicht mit dem Blitter erledigt werden, da dieser das Fast-RAM nicht ansprechen kann. Und wenn das File regulär geladen wird, dann dauert dies bei 90 kByte immerhin noch eine Sekunde. Mit einem kleinen Trick konnte das Problem doch noch gelöst werden: der Anfang des zu spielenden Sounds (die ersten 500 Byte) wird in das Chip-RAM geladen, Kriterien wie Soundlänge und Abspielgeschwindigkeit werden festgelegt, der Sound wird nun angespielt und fertig geladen. Dazu mußte die alte Soundroutine stark modifiziert werden, wie beispielsweise eine Umsetzung auf Lattice C, eine schnellere Laderoutine, die Zusammensetzung des Filenamens aus einer bestimmten Kennung und der übergebenen Soundnummer, eine Umsetzung der Laderoutine auf das von MBB verwendete Fileformat und keine Loops mehr, sondern warten bis der Sound einmal ausgegeben ist und dann den Tonkanal abschalten. Die fertige Routine mußte nun nur noch bei MBB vorgeführt und angepaßt werden. Das stellte auch den schwierigsten Teil dar, denn der Amiga mußte zum Test mit einem zweiten Task den PC simulieren und per Zufall Werte in das Janus-RAM schreiben, die dann von der Soundroutine wieder gelesen wurden.

Wurde vergessen, das Janus-RAM zu aktivieren, so wurde dies stets mit Gurus oder Abstürzen quittiert, die den eifrigen Programmierer fast in den Wahnsinn trieben. Zur Zeit wird das Steuerprogramm für den PC geschrieben, wenn das dann fertig ist, wird der Amiga den Hostrechner simulieren und Daten über die RS232 an den PC schicken, die dieser dann weiterverarbeitet. Wenn dann soweit alles funktioniert, wird der Amiga nach Hamburg verfrachtet und dort an den Hostrechner angeschlossen. Des weiteren findet der Amiga bei MBB noch ein interessantes Anwendungsgebiet, da auf ihm mit Hilfe von Sculpt 3D die Landschaft eines Panzersimulatordemos entworfen wurde. Zur Probe wurde diese Landschaft dann auf dem Amiga noch ein paarmal aus verschiedenen Perspektiven berechnet. Die fertige Landschaft wurde dann in das PC-Format übertragen.

(Andreas Fuchs/mss)

Amiga im Wunderland

Da sitzt Du da, Roger Rabbit im Herzen und Garfield im Sinn. Die Festplatte ist leer wie ein Hollywood-Pool im Winter. Sie wird sich füllen mit zappelnden Wesen, und sie alle werden Deine Story spielen: It's toon-time, Trickfilmzeit! Computer und Comics? Widersprüche? Was ist seltsamer: „Ich Tarzan, Du böse“ oder „blit buff disbuff“? Wir wollen sehen . . .



Trickfilmproduzent Lorenz Wittmann in Aktion.

Mal schrill und frech, mal schräg und komisch, dann wieder sentimental und tiefsinnig, und in seinen Sternstunden einfach bezaubernd — so war der Trickfilm zu Steamboat Willy's (erster Disney-Trickfilm) Zeiten, und so ist er auch heute noch. Er verführt den Zuschauer dazu, für eine begrenzte Zeit das zu glauben, was er sieht und sehen soll, ohne zu hinterfragen und nach den Regeln der Vernunft zu analysieren, ob das wahr, gelogen, gefährlich oder einfach nur albern ist.

Da sieht man, wie der Hauptdarsteller dreißig Stockwerke hoch mit verbundenen Augen auf dem Baugerüst eines Wolkenkratzers plötzlich ins Leere tritt. Wieso da einer derart unsinnige und mörderische Spaziergänge unternimmt, überlegt man nicht. Man zuckt zusammen. Und nichts ist dann befriedigender, als wenn der kleine einsame Tramp dem großen, dicken Widerling zum Schluß endlich das Gesicht verbeult und die Frau ausspannt.

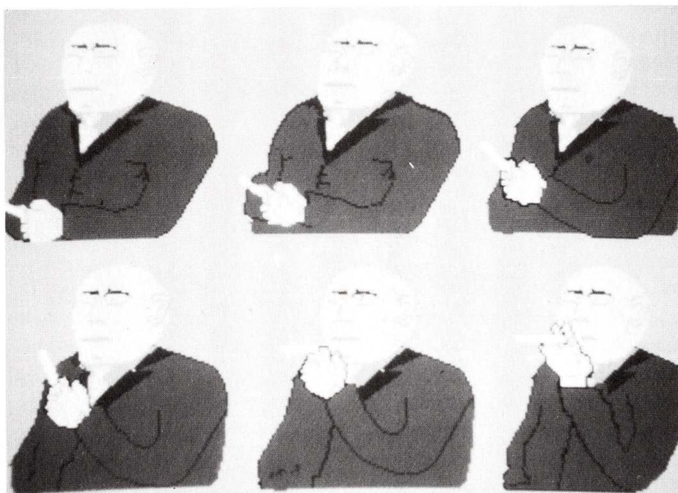
Kein anderes Genre als der Zeichentrickfilm eignet sich besser dazu, absurde und phantastische Geschichten genüßlich und in allen Farben zu schildern — jedenfalls ohne Leib und Leben der Beteiligten über Gebühr zu strapazieren. Da kann einer auf der Flucht vor der Polizei aus vollem Lauf wie ein Squashball gegen eine Mauer klatschen, abprallen („Sprooooing!!“) und den Verfolger von den Füßen fegen, um mit einem Verband um den Kopf sofort weiter und einer wütenden Bulldogge in den Rachen zu rennen. Aber so sicher wie die Kreuzer in Dagoberts Geldspeicher ist eines: Nichts ist so schwer herzustellen wie Komik und Unterhaltung. Und schon gar nichts ist so langwierig wie die Herstellung eines unterhaltsamen Zeichentrickfilms. Völlig klar also, daß die Zeichentrick-Künstler sehr schnell darauf kamen, sich bei ihrer schönen, aber eben äußerst aufwendigen Arbeit von einem auf die Bilderzeugung spezialisierten Computer helfen zu lassen. Das Interessante dabei ist, daß bei einem solchen Prozeß der Rechner völlig in den kreativen Prozeß integriert ist, stärker jedenfalls als bei einem Schriftsteller, der seine Schreibmaschine durch einen Computer und ein Textverarbeitungs-Programm ersetzt. Welche Rolle spielt also der Amiga als Bildverarbeitungsspezialist bei einer derartigen Kooperation in einem filmischen Prozeß, der so sehr von der spezifischen Sichtweise des Künstlers und seinem eigenen persönlichen Witz und Stil geprägt ist? Schiebt sich der Rechner bei der Arbeit in den Vordergrund oder fungiert er wirk-



Vorbereitung mit DPaint und Grafiktablett.



Jede Phase wird auf Folie gemalt.



Einzelphasen werden mit „Director" . . .



. . . in den feststehenden Hintergrund eingeblendet.

lich nur als eine Art Turbo-Bleistift, als Hilfsmittel oder als Kreativitätsverstärker? Um diesen spannenden Prozeß der Annäherung eines Künstlers an ein Gerät, das immerhin die Zukunft seiner Zunft mitbestimmen kann, nachzuempfinden, betreten wir neugierig das Studio eines Profis und sehen ihm bei der Arbeit an einem Trickfilm über die Schulter. Lorenz Wittmann ist solch ein Profi: Maler, Zeichner, Trickfilmer und Fotograf in einer Person, produziert er seit Jahren für die „Aktuelle Stunde" des Westdeutschen Fernsehens einen wöchentlich gesendeten Zeichentrick-Spot, in dem der Bundesadler von der Wand des Bonner Parlaments herab seine bissigen Kommentare zur Lage der Nation gibt. Lorenz Wittmann und seinen Teamgefährten, den „Floh de Cologne"-Kabarettisten und Autor Hans J. Frank, hält diese Serie nahezu jeden Tag auf Trab. In seinem Studio im Westerwald entstehen Woche für Woche 120 Sekunden Trickfilm, die gezeichnet, mit der Trickkamera aufgenommen, geschnitten und synchronisiert werden müssen, denn jeden Freitag wird im Filmhaus des WDR in Köln ein neuer Spot pünktlich zum Sendetermin fertiggestellt. Obwohl natürlich bei der Produktion einer Serie bestimmte Abläufe und Arbeitsschritte im Laufe der Zeit standardisiert werden können, sind es prinzipiell die gleichen wie bei den vielen anderen Trickfilmproduktionen, die Lorenz Wittmann in seiner künstlerischen Laufbahn abgedreht hat. Im handgezeichneten Trick- wie im Computer-Animationsfilm ersetzen also simulierte Bewegungen von Zeichnungen, Modellen oder Gegenständen die Aktionen lebendiger Darsteller. Beim Zeichentrickfilm werden die Aufnahmen von einer fest auf einer Standsäule montierten, fahrbaren und in ihren Achsen beweglichen Kamera per Einzelbildschaltung fotografiert; die jeweilige Zeichnung liegt darunter, waagerecht auf dem Tricktisch fixiert und mit ebenfalls festinstallierten Lampen gleichmäßig ausgeleuchtet. Die Szene wird Bild für Bild aufgenommen und die Lage des Objekts, das sich bewegen soll, zwischen zwei Aufnahmen minimal verändert. Bei der Wiedergabe des Filmes mit der normalen Projektionsgeschwindigkeit von 24 oder 25 Bildern in der Sekunde werden diese kleinen Verschiebungen in den Einzelbildern dank der vorzüglichen Einrichtung der Trägheit unseres Auges (der wir als Zuschauer unsere Abendunterhaltung und als Filmemacher unser täglich Brot verdanken) als kontinuierliche Bewegung wahrgenommen. Es ist ganz klar, daß für Lorenz Wittmann bald schon die Frage nach der Unterstützung seiner Arbeit durch einen Computer aktuell wurde. Es ergab sich der Kontakt zu einer Kölner Animations- und Filmproduktion, wo wiederum, angestoßen durch die Arbeit mit dem Amiga, das Thema Trickfilm auf große Gegenliebe stieß. Man legte sofort los. Schaltete den Amiga ein, stellte das Telefon ab, verwies den Hund unter den Schreibtisch und vertröstete die Freundin. Es wurde gedacht. Und dann passiert das, was eigentlich passieren muß, wenn man versucht, die ehrwürdige Kunst des Trickfilmes elektronisch zu unterwandern: Dann sitzt Du da, Rosinen im Kopf, die Maus fest in der Hand, und nichts, garnichts geht! Der Zeichner, der mit einem ordinären Bürobleistift täglich neue Welten erschafft und aufs Papier zaubert, hält plötzlich einen grauen Plastikklotz mit zwei Druck- ▶



Der fertige Trickfilm wird am Schneidetisch synchronisiert.

knöpfen darauf in der Hand, der ihm als „Maus" vorgestellt wurde. Damit, beziehungsweise mit einem kleinen, leicht angeschmutzten Ball, der unten aus dem Plastikklotz herauslugt, soll er irgendwo auf dem Schreibtisch herumfahren, so daß ein kleiner Pfeil auf dem Monitor eine Spur hinterläßt — aber nur, wenn er gleichzeitig auf einen der beiden Druckknöpfe auf der Maus drückt oder auch nicht, je nachdem, ob er eine Spur zeichnen oder mit der Hintergrundfarbe übermalen — also ausradieren — will. Dabei schaut er eisern geradeaus und bahnt sich mit der Maus einen Weg durch Kaffeetassen und Diskettenboxen. An diesem Punkt aber hatte die Liebe zur Technik ein Ende. Wie kann man von einem intelligenten und bereitwilligen Menschen, der seinen Lebensunterhalt lange und gut damit verdient hat, andere Menschen mit seiner Kunst zu erfreuen, ernsthaft verlangen, sich auf einen Plastikklotz einzustellen, der eigentlich nur vier Pfeiltasten auf einer Computertastatur ersetzen soll, mit denen man Bildschirmschablonen zum Zwecke der Datenerfassung ansteuert?

Die gewohnte, notwendige und lang trainierte Koordination zwischen der zeichnenden Hand und dem kontrollierenden Auge, die Basis jeder noch so einfachen Zeichnung, fehlt. Anstatt auf dem Papier, unmittelbar neben der zeichnenden Hand sehen zu können, wie Linie für Linie ein Bild entsteht, versucht das Auge ständig, den Bezug herzustellen zu dem, was die Hand tut und was der Bildschirm anzeigt. Es ist nun fast unmöglich, eine im Kopf vorhandene Idee klar und einfach auszudrücken. Natürlich kann man diese Diskrepanz wegtrainieren. Aber wie wirkt sich diese Umerziehung auf etwas so Substantielles wie den individuellen und bekannten Strich eines professionellen Zeichners oder Grafikers aus? Glücklicherweise gibt es für den Amiga Zeichentabletts, die ihren Zweck einigermaßen erfüllen. Hier trifft das seltsame Wort „Schnittstelle" den Kern der Sache: Ohne Zeichentablett oder andere denkbare und sinnvolle Eingabehilfen ist jeder, der den Computer zum Bildermachen nutzen will, physisch buchstäblich von seiner Arbeit abgeschnitten.

Nachdem also mit Hilfe des Zeichentabletts dieses scheinbar profane, aber natürlich grundlegende Hindernis auf dem Weg zum computerunterstützten Trickfilm genommen ist, zeigen sich langsam die Vorteile der Arbeit mit dem Amiga: Die Hand des Zeichners kann wie gewohnt mit Papier und Bleistift arbeiten, und die Männlein und Weiblein aus Linien und Flächen entstehen immer schneller. DPaint II zeigt sich dabei ganz von seiner besten Seite: Das Programm steht dem Künstler nicht im Weg und beansprucht zuviel Aufmerksamkeit, sondern unterstützt ihn bei seiner Arbeit. Farben lassen sich mühelos in immer gleichem Mischungsverhältnis herstellen, mit den Brushes kann man die Figuren leicht neu positionieren, bei der Perspektivzeichnung von Hintergründen gibt das Programm Hilfestellung. Wenn jetzt noch das Zeichentablett strichsensitiv wäre, also die Strichbreite je nach Druck des Stiftes variierte, wäre es fast zu schön.

Blättern in Gesichtern

DPaint merkt man an, daß es von Praktikern programmiert wurde, und darin liegt auch das Geheimnis seines Erfolges, aber dieses Prinzip scheint sich noch nicht ganz durchgesetzt zu haben. Gerade für den Trickfilmer ist es wichtig, zwei aufeinanderfolgende Phasen, die Detailbewegung eines Gesichts zum Beispiel, direkt hintereinander prüfen zu können, um die Veränderung zwischen den Bildern zu korrigieren. Bei der rein manuellen Arbeit blättert er dazu schnell zwischen den beiden Seiten hin und her. Genau diese Funktion simulieren bei DPaint die beiden unabhängigen Speicherbereiche, die jeweils ein Bild darstellen, und zwischen denen man per Tastendruck hin- und herspringen kann.

Technische Voraussetzungen allein reichen jedoch nicht aus, um sich Animationszeichner zu nennen. Natürlich muß man zeichnen können, sich auf seinen individuellen Strich konzentrieren, ihn ausbauen können, seine Ausdrucksfähigkeit schulen. Aber man muß noch etwas beherrschen: Man muß sehen, beobachten können. Dabei wird man feststellen, daß jede Bewegung, sei es das Aufstehen von einem Stuhl oder der Flug einer Rakete um den Mond, bestimmte Abschnitte aufweist, die einen Moment relativen Stillstands darstellen und zwischen denen die Aktion gleichmäßig weitergeht. Diese Handlungsabschnitte von einer Hauptstellung zur nächsten werden als Phase bezeichnet.

Dazwischen liegen die Nebenphasen, die natürlich auch gezeichnet werden müssen. Dabei kann man, wenn man sich nicht anders zu helfen weiß, ganz mechanisch vorgehen. Man verbindet einfach möglichst viele Punkte, beispielsweise die Schultergelenke, in beiden Hauptphasen miteinander und teilt diese Linie durch die Anzahl der Zwischenphasen. Wenn man das mit ausreichend vielen Punkten macht, ergeben sich die Zwischenphasen von selbst, allerdings linear und nicht sehr pfiffig. Merken Sie was? Genauso arbeiten die Programme, mit denen man Animationen auf dem Amiga herstellen kann, nämlich mit der Berechnung der Zwischenphasen zwischen zwei oder mehreren Hauptphasen oder Schlüsselszenen, den keyframes.

Auch ohne allzu tief in dieses Thema einzudringen, wird doch eins klar: An dieser Stelle treffen sich die beiden Arbeitsweisen, die des Trickfilm-Zeichners und die des Computers, relativ zwanglos. Diese Einzelphasen-Berechnung ist tatsächlich das, wonach unser Profi Lorenz schon längere Zeit sucht; sie würde es ihm erlauben, die Hauptphasen, die für den Rhythmus der Geschichte wichtig sind, ganz nach Belieben zu zeichnen, und dem Kollegen Computer die Interpolation der Zwischenphasen zu überlassen. Diese Zwischenberechnungen müssen durchaus nicht nur linear angelegt sein; der Weg, den der einzelne Punkt zwischen den keyframes zurücklegt, kann durch eine frei gezeichnete Kurve markiert sein oder vom Computer durch ein mathematisches Konstrukt wie ein spline (eine weiche Kurve, die mehrere Punkte verbindet) vorgeschlagen werden. Es existieren Softwarepakete wie Animation Apprentice, das bei genügend Arbeitsaufwand recht ansehnliche dreidimensionale Puppentrickfilme herstellen kann, ein einfaches Programm wie Fantavision, das für animierte Strichzeichnungen durchaus brauchbar ist, oder ähnliche wie der Movie Setter. Anschließend kann man den fertigen Trickfilm innerhalb gewisser Grenzen weiter bearbeiten. Zum Beispiel mit den Blitter-Befehlen des Programms Direktor kann man Hintergründe nachträglich unterlegen, die Farbpalette des Werkes verändern, Blenden einfügen und ähnliches. Auch die neuen Programme aus der Animation Apprentice-Serie, Multiplane, Editor, Stands, Effects oder Videola! von Aegis ermöglichen die Nachbearbeitung einer Animation. Der Trend geht also zur integrierten digitalen Produktion von Trickfilmen; Bedarf ist sicher vorhanden.

Wenn es dem intuitiv veranlagten, kreativen Menschen gelingt, sich des algorithmischen, reduzierenden und zweckgerichteten Werkzeugs Computer ohne jeden Verlust und Verzicht zu bedienen, wird es zu einer Bereicherung kommen. Keine technische Erfindung kann so vernünftig und so sinnvoll sein, daß der Mensch nicht damit genau das Maß an Unsinn verbreiten kann, welches er zur Erholung von derselben bitter nötig hat. Zwar sagt ein Bild mehr als tausend Worte, aber noch sehr viel mehr sagt eine Sahnetorte im Gesicht des Helden . . .

(Manfred Heinze)

Revolution im BASIC-Reich

Allen Kritiken zum Trotz ist BASIC die verbreitetste Programmiersprache. Was an Fähigkeiten im Vergleich zu C und Assembler fehlte, erschließt sich den Hobby-Programmierern mit dem neuen GFA-BASIC.

Spezialisten, die mit komplexen Hochsprachen arbeiten, bemängeln an BASIC, daß es den Fähigkeiten des Rechners nicht gerecht werde und zum unstrukturierten Programmieren verleite. Aber schon mit dem Amiga-BASIC lassen sich bei entsprechendem Know-how viele Features verwirklichen, ohne dabei Spaghetti-Code zu produzieren.
Technische Lücken des BASIC stopft nun das GFA-BASIC der Düsseldorfer Firma GFA-Systemtechnik. Auf dem Atari hat das GFA-BASIC bereits Maßstäbe gesetzt. Jetzt öffnet sich dem Amiga-User die Düsseldorfer Schatztruhe mit neuen Möglichkeiten der Programmierung.
Erste Annehmlichkeiten fallen bei Arbeitsbeginn auf. Die Editierung von Unterprogrammen wird durch die Möglichkeit, diese durch Tastendruck aufzuklappen, erleichtert. So entfällt das langwierige Scrollen über endlose Bildschirmseiten. Hier wird deutlich, daß der Editor auf Programmiertexte spezialisiert ist. Er überprüft die Richtigkeit der Syntax einer Zeile sofort nach Bestätigung. Dies erweist sich als sehr nützlich. So wird man nicht erst im Programmablauf auf Fehler hingewiesen. Auch das automatische Einrücken von Programmzeilen in Bedingungen oder Schleifen führt der Editor sofort aus. So gewinnt ein Programm an Übersichtlichkeit und fördert die strukturierte Programmierung. Der Programmierer wird zur klar gegliederten Programmierung gezwungen. Pro Programmzeile ist nur eine Befehlsausführung erlaubt. Das mag enorme Platzverschwendung sein, bietet aber den Vorteil großer Übersichtlichkeit.

Edit-Funktionen ohne Schlaftabletten

Blockfunktionen, wie Kopieren, Einsetzen, Verschieben, Löschen, sogar das Abspeichern eines Programmteils werden schnell ausgeführt. Ein markierter und kopierter Programm-Block erscheint schneller auf dem Bildschirm, als das Auge sehen kann. Er baut sich nicht erst zeilenweise auf, wie das beim AmigaBASIC der Fall ist. Unterprogramme bestehen beim GFA-BASIC aus Prozeduren, die mit RETURN, ENDSUB, ENDPROC oder einzeiligen und mehrzeiligen Funktionen beendet werden. Funktionen werden durch FUNCTION deklariert und mit ENDFUNC verlassen. Nahezu gewaltig ist die Anzahl der zur Verfügung stehenden Befehle. Besonders herausragend sind Befehle zur Fallunterscheidung ▶

und Schleifenverwaltung. REPEAT und UNTIL zum Beispiel bilden eine nichtabweisende Schleife. Der Vorteil besteht darin, daß diese Art der Schleife mindestens einmal durchlaufen wird. Hierdurch wird vermieden, daß eine Schleife in einem Programm ignoriert wird. Aber auch DO und LOOP, WHILE und WEND bilden durch die verschiedensten Verknüpfungen viele Möglichkeiten, individuelle Schleifenarten zu konstruieren (DO WHILE, DO UNTIL, LOOP WHILE). Mit EXITIF können alle diese Schleifentypen verlassen werden. Auch die SELECT-Anweisung dient in Zusammenhang mit CASE und ENDSELECT der Fallunterscheidung. Ein Beispiel der Joystickabfrage für SELECT-CASE:

```
OPENW 0
DO
  joystick=STICK(1)
  LOCATE 10,10
  SELECT joystick
  CASE 1
    PRINT "Oben   "
  CASE 2
    PRINT "Unten "
  CASE 4
    PRINT "Links "
  CASE 8
    PRINT "Rechts"
  DEFAULT
    PRINT "Mitte "
  ENDSELECT
LOOP
```

Das DEFAULT am Schluß tritt ein, wenn keiner der CASE-Fälle eintritt. Wie bei ELSE-IF lassen sich Mehrfachverzweigungen anwenden. Zu den zahlreichen Befehlen für die Schleifenbearbeitung und Fallunterscheidung sind aber auch eine ganze Reihe von Befehlen hinzugekommen, welche die String- und Variablenbearbeitung erleichtern. FORM INPUT erlaubt zum Beispiel die Definition eines Eingabestrings und dessen maximale Länge. Hierdurch kann vermieden werden, daß eine zu lange Eingabe den Bildaufbau zerstört.

```
b$="10.02.89"
FORM INPUT 8 as b$
```

läßt eine maximale Eingabe von acht Werten zu. Bei RETURN wird der Wert b$ einfach übernommen. Das Sortieren von Feldinhalten ist nun auch mit den Befehlen QSORT und SSORT möglich. Die Inhalte eines Arrays lassen sich entweder nach dem Quicksort- oder Shellsort-Verfahren sortieren.

```
DIM name$(10)
OPENW 0
FOR n=1 TO 10
  LINE INPUT name$(n)
NEXT n
PRINT "Namen unsortiert"
FOR n=1 TO 10
  PRINT name$(n)
NEXT n
QSORT name$()
LOCATE 20,12
PRINT "Namen sortiert"
FOR n=1 TO 10
  LOCATE 30,n+12
  PRINT name$(n)
NEXT n
```

Dieses Beispiel sortiert im Quicksort-Verfahren das Feld name$(x). Das ist besonders wertvoll für Dateiverwaltungen. Feldinhalte lassen sich mit INSERT neu einsetzen und durch DELETE löschen. ARRAYFILL füllt ein Feld mit definierten Werten. Auch Befehle zur Bitmanipulation, die das Neusetzen, Negieren und Löschen eines Bits erlauben, fehlen nicht. Das GFA-BASIC unterstützt nun auch alle darstellbaren Bildschirmmodi. Der Halfbrite-Modus mit 64 Farben ist ebenso selbstverständlich wie der HAM-Modus mit seinen 4096 gleichzeitig darstellbaren Farben; maximal 6 Bitplanes können verwaltet werden. Die Grafikbefehle wie DRAW, DRAW TO, BOX, PBOX, POLYLINE, POLYFILL, ELLIPSE und PELLIPSE bieten neben den Standard-Grafikbefehlen wie CIRCLE, LINE POINT und PLOT vielseitige Möglichkeiten. Auch Füllmuster lassen sich mit DEFFILL definieren. Besonders interessant ist die Möglichkeit, Logoähnliche Befehle anzuwenden:

```
OPENW 0
SETDRAW 250,70,0
DRAW "rt 90 fd 100 rt 90
      fd 50 rt 90 fd 100
      rt 90 fd 50"
```

bewegt einen imaginären Stift auf dem Bildschirm und zeichnet ein Quadrat. Die Sprites und BOBs werden ebenfalls vom GFA-BASIC unterstützt. Auch sie lassen sich unter GFA-BASIC einfach manipulieren und bewegen. BOBs können mit OBJECT.SHAPE in einem String abgelegt werden. Die Beschleunigung, Geschwindigkeit und Kollision eines BOBs mit anderen kann problemlos definiert werden. Leider konnte anhand der vorliegenden Testversion nicht nachgeprüft werden, ob das „Flimmern“ von größeren BOBs nun beseitigt ist. Auch die Verwaltung von Pull-Down-Menüs bietet unter GFA-BASIC vielfältige Möglichkeiten. MENU KEY zum Beispiel weist einem Menüeintrag einen ASCII-Code zu. Nach dieser Anweisung kann die Menüauswahl auch durch Drücken der definierten Taste mit der Amiga-Taste erfolgen. Doch nicht nur das. Menüeinträge können nach Anwahl durch ein Häckchen abgehakt werden, aber auch invertiert, mit einem Kasten umrandet oder invers dargestellt werden. Die hauptsächlichen Befehle zur Menü-Verwaltung sind ON MENU GOSUB, ON MENU KEY, ON MENU BUTTON, welcher die Maus abfragt, und ON MESSAGE GOSUB.

```
OPENW 0
DIM nam$(10)
DATA Menue1, Laden
 Speichern
Löschen        ,
FOR n=0 TO 4
  READ nam$(n)
NEXT n
MENU nam$()
MENU KEY 1,65
MENU KEY 2,66
MENU KEY 3,67
ON MENU GOSUB auswertung
DO
  SLEEP
LOOP
PROCEDURE auswertung
  m=MENU(0)
  wert%=(SHR(MENU(2),5)
  AND 63)+1
  PRINT nam$(wert%)
  PRINT wert%
RETURN
```

Außerdem unterstützt GFA-BASIC die Einrichtung von Requestern und Gadgets.

Dieses Beispiel definiert ein Alert-Requester, welches sich problemlos auswerten läßt. Der FILESELECT-Befehl erzeugt eine Datei-Auswahl-Box. Auch diese läßt sich individuell definieren. Das Erstellen von Eingabemasken wird durch das „R.C.T.“, das „Requester-Construction-Tool“, erleichtert. Das „R.C.T.“ ist jedoch nicht im Lieferumfang des GFA-BASIC enthalten. Auch die Aufrufe von Systemroutinen sind problemlos durch RCALL möglich. Erleichtert wird dies dadurch, daß die dazu notwendigen Funktionen „fest“ im Interpreter eingebunden sind. Die vom Amiga-BASIC bekannten fd.- und .bmap-Files können nun getrost in den Abfalleimer befördert werden. Assembler- und C-Programme lassen sich mit C: oder CALL aufrufen. Außergewöhnlich ist auch die Möglichkeit, Speicher zu reservieren und wieder freizugeben. Über den Befehl MALLOC kann Speicher reserviert, durch MFREE wieder freigegeben werden. Außerdem kann mit dem Befehl INLINE Speicherplatz innerhalb des BASIC-Programms reserviert werden. Der Sinn besteht darin, Maschinenroutinen in das BASIC-Programm einzubinden und dieses buchstäblich mit abzuspeichern. So sind die eingebundenen Routinen sofort nach dem Laden vorhanden. Das GFA-BASIC hat schon auf dem Atari ST Zeichen gesetzt. Mit der Umsetzung auf den Amiga ist den Entwicklern des GFA-BASIC ein Wurf gelungen, der auch auf dem Amiga für Abwechslung sorgt.

(Eric Spangenberg)

```
OPENW 0
ALERT 0," Wie geht es Ihnen
heute?",2,"Schlecht| Mittelmäßig |Gut",a%
LOCATE 10,10
SELECT a%
CASE 1
  PRINT "Das ist bedauerlich!"
CASE 2
  PRINT "Was soll's!"
CASE 3
  PRINT " Das ist erfreulich!"
ENDSELECT
```

Videotexten mit Amiga

Mit Videotext können aus den Austastlücken des Fernsehbildes aktuelle Nachrichten, Sportberichte, Börsenkurse, Wetterinformationen und vieles mehr aufgerufen werden. Teletext ist ein Informationsservice, der bisher nur mit einem dafür ausgerüsteten Fernsehgerät empfangen werden konnte. Nun ist es auch möglich, mit Amiga und dem Teletext-Decoder TTD 100 diese Informationen zu entschlüsseln.

Ein Fernsehsignal liefert mehr Zeilen als am Bildschirm angezeigt werden, denn die ersten 22 Zeilen, die im Fernsehsignal enthalten sind, werden am Bildschirm nicht angezeigt. Davon können 16 für Videotext verwendet werden. Derzeit werden sechs bis sieben Zeilen genutzt. Vom Sender werden die Daten digital mit einer Übertragungsrate von 6,9375 Megabits pro Sekunde gesendet. Angeschlossen wird der TTD 100 an die parallele Schnittstelle des Amiga. Das zu dekodierende Videosignal wird über eine Chinchbuchse in den Videotext-Decoder eingespeist. Benötigt wird ein 1-Vss-Composit-Video-Signal (BAS oder FBAS). Das Videosignal kann entweder vom Videoausgang eines Fernsehgerätes oder Videorecorders oder von einem Tuner stammen. Nach dem Anschluß von Signalquelle und Decoder an den Amiga kann die Software geladen werden. Die zur Hardware mitgelieferte Programmdiskette enthält das eigentliche Decoderprogramm VTEXT. Wenn das Programm geladen wurde, erscheint am Bildschirm „Page #100“, die sogenannte Indexseite. Nach einer maximalen Abtastzeit von 60 Sekunden sollte der Bildschirm fertig aufgebaut sein. Die Abtastzeit ist davon abhängig in welchem Kanal sich das System gerade befindet. Wenn kein Bild erscheint, wurde kein Teletext-Signal gefunden.

Weil der Amiga ja primär ein Datenverarbeitungsgerät ist, kann er zusammen mit dem Decoder und der menügesteuerten Software viel mehr als ein Fernsehempfänger. Die Seiten werden zwischengespeichert und man kann vor- und zurückblättern. Bildschirmseiten können in verschiedenen Formaten auf Diskette gespeichert werden. Ferner sind auch Hardcopies als Text und Grafik möglich. Hier das ganze Menü:

PROJEKT Laden
Laden einer Videotext-Seite, die im Videotext-Format auf Diskette gespeichert wurde.

PROJEKT Speichern VT
Die am Bildschirm angezeigte Seite wird im Videotext-Format auf Diskette gespeichert.

PROJEKT Speichern ASCII
Die am Bildschirm angezeigte Seite wird im ASCII-Format (als Text) auf Diskette übertragen. So gespeicherte Seiten können in ein Daten- oder Textverarbeitungsprogramm eingebunden werden. Dazu eine Anwendung, die bereits von Anlegern praktiziert wird: Die aktuellen Börsenkurse werden als ASCII-File eingelesen und in ein Börsenprogramm übertragen. Dieses vergleicht die Kurse und gibt alle Abweichungen aus.

PROJEKT Speichern IFF
Die am Bildschirm angezeigte Seite wird im IFF-Format (als Bild) auf Diskette gespeichert und kann in Zeichen- oder DTP-Programme eingebunden werden.

PROJEKT Info
Die Programmversion und das aktuelle Chip- und Fast-Memory werden ausgegeben.

DRUCKEN Grafik
Die sichtbare Bildschirmseite wird als Grafikbild ausgedruckt. Voraussetzung ist natürlich, daß vorher der Drucker mit Preferences richtig eingestellt wurde.

DRUCKEN ASCII
Die sichtbare Bildschirmseite wird als Text ausgedruckt.

ANZEIGE Stop
Manche Videotext-Seiten bestehen aus mehreren Bildschirmseiten, die nach der Auswahl in einer Endlosschleife angezeigt werden. Mit diesem Menüpunkt kann die Anzeige angehalten werden.

ANZEIGE Hilfe
Es gibt Videotext-Seiten, die eine Antwort zu einer Frage verstecken. Mit diesem Menüpunkt kann die Antwort sichtbar gemacht werden.
Es wurden alle Varianten der Einspeisung des Videosignals ausprobiert. Es wurden auch verschiedene Empfangslagen getestet. Hierbei zeigt sich, daß sogar ein verrauschtes HF-Signal noch einwandfrei empfangen wurde. Auch schwache Geisterbilder störten nicht. Schlecht übertragene Zeichen werden erkannt und vom Decoder durch Leerzeichen ersetzt. Dadurch werden Übertragungsfehler bei sehr schlechtem Empfang soweit als möglich unterdrückt. Der Videotext-Decoder TTD 100 ist nicht nur als Ergänzung für Fernsehgeräte ohne Videotext zu empfehlen, sondern wegen der vielen zusätzlichen Möglichkeiten eine sinnvolle Erweiterung Videotext-tauglicher Geräte.

Lieferumfang:
- Hardware TTD 100 mit Anschlußkabel
- Programmdiskette 3½ Zoll
- Bedienungsanleitung

(rw)

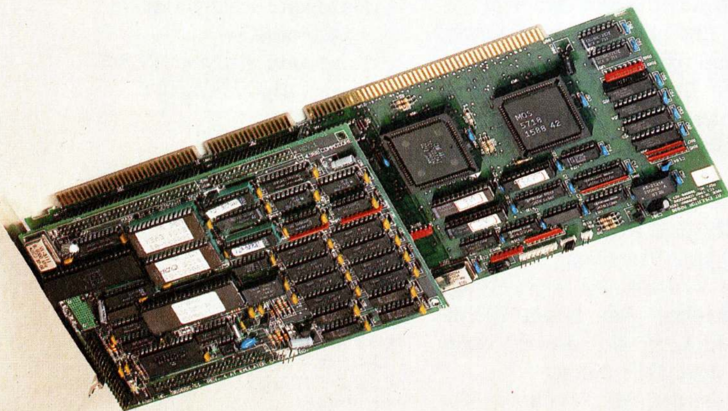
Info:
PRINT-TECHNIK München, 8000 München, Nikolaistraße 2, Tel.: 0 89/36 81 97,
PRINT-TECHNIK Wien, A-1060 Wien, Stumpergasse 34—36, Tel.: 02 22/5 97 34 23,
Preis: 298 Mark.



Nicht nur die Wetterdaten aus Österreich lassen sich über die TV-Austastlücke in den Computer einlesen.

AT-Karte im Reifungsprozeß

Nach zweijähriger Ankündigungszeit zeichnet sich endlich die Markteinführung eines IBM-AT-kompatiblen Bridgeboards ab. Viel ist bisher angekündigt und berichtet worden. Jetzt erst sind Daten bekannt, die über die Gerüchteküche hinausgehen.



Im kombinierten Amiga-PC-Slot entwickelt die AT-Karte ihre bis zu 10 MHz schnelle Leistung. 80286 oder 80287 werden automatisch konfiguriert. Das Board wird mit MS-DOS 3.2 ausgeliefert, ist jedoch auch OS/2-fähig.

Im Oktober letzten Jahres waren auf der Orgatechnik in Köln die ersten verkaufsfähigen AT-Karten der Öffentlichkeit vorgestellt worden. Mittlerweile sind Veränderungen bekannt, die diese Version als Prototyp erscheinen lassen. Die zweite europäische Entwicklerkonferenz für den Amiga in Frankfurt bot jetzt den geeigneten Rahmen, die AT-Karte einem genaueren Test zu unterziehen.
Der Prozessor der Karte ist ein Intel 80286, der mit 8 Megahertz getaktet wird (eine 10-Megahertz-Taktung ist für die endgültige Version im Gespräch). Der Steckplatz für einen 80287-Mathecoprozessor ist vorhanden. Der Hauptspeicher auf der Karte ist ein Megabyte groß.

Die gesamte Hardware ließ sich jedoch nicht auf einer einzigen Platine unterbringen, so daß eine Huckepackplatine auf die Steckkarte aufgesetzt ist. Zu der Karte wird ein 5¼-Zoll-Diskettenlaufwerk geliefert, das sowohl die 360-Kilobyte-Disketten lesen kann als auch 1,2-Megabyte-Disketten.

MS-DOS 4.1 soll lauffähig sein

Als Betriebssystem wird MS-DOS 3.3 in Verbindung mit einer deutlich überarbeiteten Janus-Library ausgeliefert. Eines der neuen Features ist der Maustreiber, der über Amiga-Maus eine Microsoft-PC-Maus softwaremäßig emuliert. Sobald die neue, hoffentlich fehlerfreiere, Version 4.1 des MS-DOS auf dem Markt ist, wird auch diese Version für das Bridgeboard erhältlich sein. Benutzer, die genug RAM (sechs oder mehr Megabyte) auf Erweiterungskarten für den PC haben, werden OS/2, das neue Multitaskingbetriebssystem für ATs, auf der AT-Karte betreiben können. Ein Problem ist, daß laut IBM OS/2 nur mit neu gekauften Rechnern der PS/2-Serie oder von Herstellern wie Compaq vertrieben werden soll.

Die neue Janus-Library ermöglicht eine effektivere beiderseitige Verwendung der Disketten- oder Festplattenlaufwerke von der Amiga- oder der PC-Seite aus. PC-Festplatten können wie vorher auch in Amiga- und PC-Partitionen unterteilt werden; neu ist, daß jetzt der PC auch von AmigaDOS-Festplatten aus gebootet werden kann. Zu diesem Zweck kann auf Amigaseite ein beliebiger File gemountet werden, der eine PC-Festplatte emuliert. Danach kann nach dem Installieren von PCDisk der PC gebootet werden. Die neue Janus-Software arbeitet sowohl mit der AT-Karte als auch mit der XT-Karte und wird nach endgültiger Fertigstellung der Software mit beiden Produkten ausgeliefert.

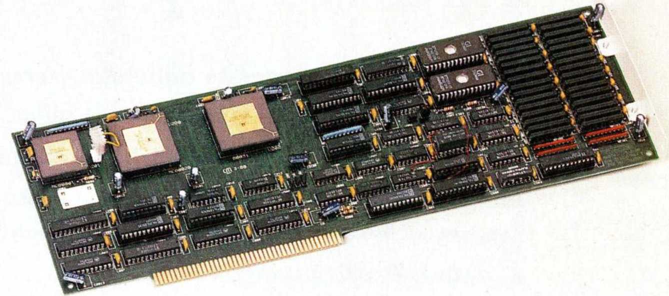
Wie schon bei der alten XT-Karte wird auf dem Amiga bisher nur MGA- oder CGA-Grafik emuliert. Höhere Auflösungen benötigen auch weiterhin eine in einem PC-Slot installierte Grafikkarte. Die Ablaufgeschwindigkeit der AT-Karte liegt gleichauf mit einem AT-Kompatiblen. Die Geschwindigkeit der Software-Emulation der Grafik hängt direkt von der Anzahl der auf dem Amiga laufenden Prozesse ab.

Einer der Hauptvorteile der Bridgeboards ist die Platzersparnis, da der Benutzer nur ein Gerät auf dem Schreibtisch stehen hat. Außerdem ist es nur mit dem Bridgeboardkonzept möglich, komfortabel Daten zwischen dem PC und dem Amiga zu übertragen. Die neue AT-Karte bietet vor allem Usern, die außer dem Amiga auch einen AT benötigen, eine preisgünstige Alternative zu vielen AT-Kompatiblen.

(Wilfried Häring)

UNIX-Power auf Amiga

Neben dem AmigaDOS gibt es jetzt UNIX für die neue 68020-Karte, das im Profibereich für Mehrplatzrechnersysteme eingesetzt wird.



Das Powerboard enthält einen Motorola-68020-Prozessor, der mit 14,3 MHz taktet. Ferner: 68881-Coprozessor (optional 68882), 68851 MMU mit 14 MHz (optional 20 oder 25 MHz) und 2 bis 4 MB 32-Bit-ROM.

UNIX ist weit mehr als ein Multitaskingsystem. Das Besondere an diesem Betriebssystem ist die Fähigkeit, mehrere verschiedene Benutzer gleichzeitig zu bedienen, also Multi-Using zu unterstützen. Dabei bekommt jeder Benutzer eine eigene UserID und kann sich ein für alle anderen Benutzer unknackbar geheimes Paßwort anlegen, mit dem nur er vollen Zugriff auf alle seine Daten besitzt, alle anderen Benutzer der Maschine haben üblicherweise die Möglichkeit Dateien wie Programmquelltexte zu lesen. Das heißt aber auch, das ein normaler Benutzer nicht die Möglichkeit hat, die Daten seiner Kollegen zu verändern, oder gar Routinen des Betriebssystems zu verändern.

Eine Sonderstellung nimmt der sogenannte SuperUser ein, der die Möglichkeit hat, für die „Wartung", das Erweitern und das Aufräumen des Systems auch direkt in Systemfunktionen einzugreifen oder die Daten eines Benutzers zu löschen oder zu verändern. Das macht UNIX-Computer zu sehr datensicheren Systemen, die nur an den Stellen Schwachstellen haben, an denen Systemprogramme dem normalen Benutzer zum Erreichen eines Zwecks vorübergehend Superuserfähigkeiten einräumen müssen.

Sensationell ist der, im Verhältnis zu anderen etablierten, Grafik-unterstützenden UNIX-Computern, sehr niedrige Preis, der sich um 15 000 Mark bewegen wird. Für diese Investition bekommt der Anwender sehr viel geboten. Amiga-UNIX ist kompatibel zu AT&T UNIX System V Version 3.1, und wird dem Standard, der von AT&T und SUN Microsystems weiterentwickelt wird, hin zu System V Version 4 folgen.

Die Hardware besteht aus dem bekannten Amiga 2000 B, der neuen 68020-Prozessor-Karte 2620 mit dem Mathecoprozessor 68881, der für UNIX nötigen Page Management Unit 68581 und vier Megabyte RAM auf einer Platine, die in den Prozessorslot des Amigas eingesteckt ist. Ergänzt wird das System von dem Festplattenkontroller 2090A und einer SCSI-Festplatte mit 80 Megabyte Kapazität und einer mittleren Zugriffszeit von 19 Millisekunden, für deren Backup ein Tapestreamer-Laufwerk mitgeliefert wird, das eine Kapazität von 150 Megabyte besitzt.

Entgegen den Hoffnungen vieler User ist es nicht möglich, AmigaDOS und UNIX gleichzeitig ablaufen zu lassen, da die internen Konzepte beider Betriebssysteme zu unterschiedlich sind. Auch eine Lösung, die AmigaDOS als Prozeß unter UNIX laufen läßt, ist nicht denkbar, da vor allem die Fähigkeiten von AmigaDOS und Intuition, die auf den Customchips beruhen, nicht vollständig lauffähig blieben. Umgekehrt ist es auch nicht möglich, UNIX unter AmigaDOS laufen zu lassen, da sich viele UNIX-spezifische Fähigkeiten mit Hilfe von AmigaDOS nicht verwirklichen lassen. UNIX weist nämlich im Gegensatz zu AmigaDOS jedem Prozeß einen eigenen Speicherbereich zu, der mit Hilfe der PMMU vor dem Überschrieben werden durch andere Prozesse geschützt ist. AmigaDOS dagegen verwaltet einen einzigen großen „Haufen" Speicher, den sich alle Prozesse teilen. Zu einen der ungeliebten Guru's kommt es meist dann, wenn ein Programm durch einen Fehler Daten eines anderen Programms überschreibt. Würde nun der Amiga wegen eines fehlerhaften Amiga-Programms abstürzen, würde er UNIX mit in den Untergang reißen. Das ist für ein Betriebssystem, das von mehreren Usern gleichzeitig benutzt wird, inakzeptabel, da allen von ihnen Daten verloren gingen. Die grafische Oberfläche von UNIX ist eine völlige Neuentwicklung, die die Customchips bis an den Rand des Möglichen ausnutzt. Ähnlich wie Intuition bietet die UNIX-Oberfläche verschiedene virtuelle Screens, wobei in jedem Screen ein eigener Window-Server installiert werden kann, der gleichzeitig mit dem Commodore-eigenen und mehreren anderen lauffähig ist.

Der Window-Server von Commodore ist hauptsächlich für die Darstellung von Text optimiert und bietet phantastische Scrollgeschwindigkeiten von Text in den Windows. Das Erscheinungsbild der Fenster unterscheidet sich stark von Intuition. Die Titelzeile und die Gadgets an den Rändern von Fenstern fehlen ganz. Dafür erscheint nach Drücken der rechten Menütaste ein Pop-up-Menü an der Stelle des Mauszeigers und bietet alle Funktionen an, die die Gadgets bei Intuition übernahmen. Nach wenigen Minuten fühlt man sich auch bei diesem System heimisch. Überlappende Fenster werfen Schatten, so daß sich ihre Anordnung in der Tiefe leicht erkennen läßt.

Jeder User, der schon Erfahrungen mit UNIX-Systemen gemacht hat, wird die Fähigkeiten der Benutzeroberfläche schnell zu schätzen wissen, zumal die übliche CShell erhältlich sein wird. Allerdings werden sich verwöhnte Benutzer nicht leicht mit der niedrigen Auflösung des Amiga-Monitors abfinden, wenn sie mehr als tausend mal tausend Punkte Bildschirmauflösung gewohnt sind. Hier wird aber der neue hochauflösende Monitor 2024 mit einer Auflösung von 1008 × 1024 Punkten Abhilfe schaffen.

UNIX wird mit allen üblichen Werkzeugen einschließlich des C-Compilers ausgeliefert, den Motorola selbst für die 68000 Chips entwickelt hat. Der Compiler unterstützt den vollen ANSI-Standard, ist sehr schnell, und liefert ein sehr gutes ausführbares Programm.

Sowohl ambitionierte Hobbyisten als auch Profis können mit dem UNIX-Amiga in die Betriebssystemzukunft starten und erleben dabei einen Amiga, der durch UNIX wesentlich an Attraktivität gewonnen hat. *(Wilfried Häring)*

Datamat wird professionell

Relationale Datenbanken ohne integrierte Programmiersprache sind weit verbreitet. Jetzt sind Upgrades zu bestehenden Datenbanksystemen erschienen, die dieses Feature integriert haben. Zu dieser Kategorie gehört Datamat Professional.

Superbase professionell war die Herausforderung für Data Becker, aus ihrem Amateur Datamat einen Profi zu machen. Datamat wurde in Funktionalität und Erscheinungsbild überarbeitet und mit konkurrenzfähigen Fähigkeiten ausgestattet.

Datamat Professional bringt mit seiner farbigen grafischen Oberfläche Spaß in die Arbeit. Anwender, die Data Becker-Programme kennen, kommen spielend mit der Bedienung zurecht. Selbst Neulinge gewöhnen sich schnell an die Handhabung.

Dennoch zeigen sich einige Programmeigenarten, die die Datenlust zum Frust werden lassen. Die Programmierer hätten zugunsten einer einfacheren Erzeugung der Eingabemasken und weniger umständlicher Bedienung auf grafische Gestaltungsmätzchen verzichten sollen.

Auch das Einbinden von Bildern und das Verknüpfen von Datensätzen wird dem Anwender nicht leicht gemacht. Durch die Verspieltheit der Benutzeroberfläche leidet die Transparenz der Funktionen.

Nun aber zu den Stärken des Programms. Die Basic-ähnliche Datenbankabfragesprache ist funktionsstark und durch ihre mächtigen Kontrollstrukturen, die Sprachen wie C, Modula-2 oder Pascal entliehen sind, leistungsfähiger als der Basic-Dialekt von Superbase Professional, zumal sich durch einen integrierten Compiler ein Zwischencode erzeugen läßt, der die Ablaufgeschwindigkeit von Programmen merklich erhöht.

Gut gelöst ist das Ausdrucken von grafisch gestalteten Formularen. Die Druckqualität ist ähnlich wie bei Textverarbeitungen, die Pagesetter und Grafikdump nutzen und ist auch zur reinen Datenausgabe einer Datenbank völlig ausreichend.

Professionelle Anwender werden zu einem geradlinigen Programm ohne Schnörkel greifen, das einen ausreichenden Funktionsumfang bietet, den sie ohne Programmiererfahrung schnell zu beherrschen und zu nutzen lernen. Heimanwender werden die ansprechende Benutzeroberfläche schätzen, sich aber fragen, ob sie wirklich die Fähigkeiten einer programmierbaren Datenbank benötigen und sich dadurch der recht hohe Anschaffungspreis rechtfertigt.

Ferner ist zu bedenken, daß der Lernaufwand zur effektiven Nutzung einer Datenbankabfragesprache ebenso hoch ist wie zum Erlernen einer universellen Programmiersprache. Programmierwillige User werden sich fragen, ob sie sich lieber mit Pascal, C oder Modula-2 beschäftigen.

(W. Häring)

Info: Data Becker Düsseldorf

Aegis Draw — dritter Anlauf

Aegis Draw und Aegis Draw Plus haben einen verbesserten und erweiterten Nachfolger: Aegis Draw2000 ist an den mathematischen Coprozessor angepaßt worden.

Anregungen engagierter CAD-Anwender haben die Aegis-Programmierer bei der Arbeit an Aegis Draw2000 beflügelt. Manches lag mit den bisherigen Draw-Programmen im argen. Die Datenkompatibilität ließ zu wünschen übrig, das Handbuch war schwer verständlich und die Zeichengeschwindigkeit war nicht gerade berauschend. Im jetzt erhältlichen Programmpaket sind solche Mängel weitgehend behoben worden. In der Verpackung liegen nun drei anstatt bisher zwei Disketten; die dritte ist die „Draw2000 Math Coprozessor Version“, die speziell für 6820/881-Benutzer gedacht ist. Für das Gros der Anwender sind zusätzliche Treiber für Plotter integriert worden. Ferner wurde die Zeichengeschwindigkeit in geringem Maße erhöht. Mager sind die Veränderungen im Menü; hinzu kam lediglich ein einziger Menüeintrag, über den sich Draw2000 direkt auf der Workbench nutzen läßt. Eine wesentliche Verbesserung wurde bezüglich der Datenkompatibilität vorgenommen. Jetzt lassen sich Dateien des Videoscape-Editors Modeler-3D lesen und



Typisches Anwendungsbeispiel für Aegis Draw: Mit Videoscape wurden IFF-Grafiken mit Vectorgrafiken zu einer Animationssequenz verschmolzen.

schreiben. Dieser Vorteil ist allerdings weitgehend den Videoescape-Fähigkeiten zuzuschreiben, da dieses Programm Daten im Draw2000-Format ablegt. Ferner können Daten mit dem DTP-Programm Professional Page ausgetauscht werden. Dabei können Draw2000-Dateien nicht nur in der maximalen Bildschirmauflösung ausgegeben werden, sondern auch in der höchsten Auflösung des jeweiligen Druckers.

Die gravierendste Verbesserung leistete Aegis am Handbuch. Es ist vollkommen überarbeitet, ergänzt und in einer verständlichen Sprache abgefaßt worden. Erstmals enthält das Buch Dokumentation zum Datenformat der Draw2000-Dateien, eine Tatsache, die PD-Programmierer zu neuen Hilfsprogrammen anregen dürfte.

Programmierfreudige werden beim Erstellen eigener Plotter-Treiber angeleitet; selbst extravaganteste Plotter sind berücksichtigt. Neben all diesen zusätzlichen Themen enthält das Handbuch alle Informationen, die im Draw Plus-Handbuch geboten sind. Allerdings wurde es vollkommen neu geschrieben und zudem sinnvoll strukturiert. Es wird jedoch nicht eingegangen auf explizite Unterschiede zwischen Draw Plus und der gestylten Version Draw2000.

Jetzt lautet die Frage: Kaufen oder nicht kaufen? Die Entscheidung liegt auf der Hand. Enthusiastische CAD-Heimanwender, die sich mit mäßigem Bedienungskomfort und gemessener Zeichengeschwindigkeit begnügen, sind mit Draw2000 gut beraten. Die Erwartungen eines Profis werden jedoch nicht erfüllt.

(O. Röhrig)

Info: Atlantis, 5030 Hürth

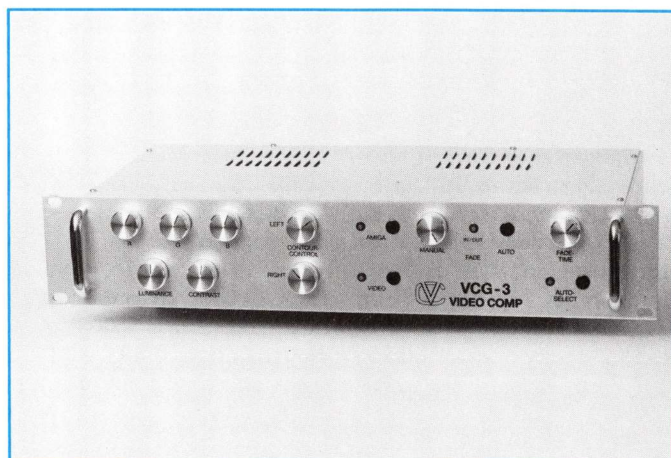
Eine Verbindung zu Video

Das Genlock Interface VCG-3 ermöglicht den professionellen Einsatz des Amiga am Videoschnittplatz oder zu Hause.

Desktop Video ist das Zauberwort für Videoamateure und Semi-Profis. Jetzt gibt es Geräte die zu einem vernünftigen Preis die Verbindung zwischen Computer und Videogeräten herstellen. Genlock Interface ist die Bezeichnung für diese Kombination aus RGB-Wandler und Videosignalmischer. Zwischen 500 und 5000 Mark muß man investieren, um seinen Amiga als Schriftgenerator, Blue Box, Retouchiergerät oder Animationsstudio einsetzen zu können.

Unser Testgenlock ist das VCG-3 der Frankfurter Firma Videocomp. Mit einem Preis von knapp 2000 Mark ist dieses Gerät sowohl für Hobbyfilmer als auch für Profis interessant. Das VCG-3 ist eigentlich für den Einsatz am U-Matic Schnittplatz gedacht (und dort getestet worden), aber auch die Verwendung am Betacam- (durchgemessen) oder VHS-Schnittplatz dürfte problemlos möglich sein.

Überhaupt ist der Umgang mit dem VCG-3 sehr unkompliziert. Ist das Gerät erst einmal richtig verkabelt (in ausgeschaltetem Zustand natürlich) wird das entsprechende Amiga-Bild, nach dem Einschalten der „Insert Workbench Hand", auf dem Rekorder Monitor sichtbar. Die weiße Hintergrundfarbe, wie auf dem Computermonitor zu sehen, wird ausgestanzt. Anstelle der Hand läßt sich jedes Objekt und jede Schrift mit dem Bild der Videoquelle kombinieren. Dies funktioniert nach dem gleichen Prinzip wie die Blue Box in der Tagesschau. Das Genlock bekommt dabei zwei Signale (Videoquelle und Computersignal) und mischt diese zusammen. Dabei wird der Hintergrund des Computersignals durch das Bild der Videoquelle ersetzt und mit dem so übrig gebliebenen Computerobjekt überlagert. Ob man dabei nun Objekte bewegt, einblendet oder schlichtweg stehen läßt spielt eine untergeordnete Rolle. Zu beachten ist dabei nur, daß die zu bewegenden Gegenstände nicht zu klein werden, denn hier könnten Probleme auftreten. Ist ein Objekt, das als Brush in Dpaint II eingespeichert ist und per Mauszeiger schnell über den Bildschirm bewegt werden soll, zu winzig, kommt es vor, daß anstelle der wenigen Pixel ganze Zeilen schwarz werden. Das sieht dann aus als ob man jede Menge „Drop Outs" auf dem Band



5,5 Megahertz Bandbreite machen den VCG-3 sogar für Profis interessant

hätte. Sobald die bewegten Objekte eine gewisse Größe (doppelter Mauszeiger) überschreiten tritt diese Schwierigkeit nicht mehr auf. Bei stehenden Objekten und Schriften gibt es überhaupt keine Probleme. Mühelos wird jegliche Schrift eingestanzt und kann mittels der verschiedenen Kontrollregler in Farbe, Luminanz, Kontrast und Kontur verändert werden. Für Liebhaber sanfter Ein- und Ausblendungen bietet das VCG-3 zwei Optionen. Entweder manuell oder automatisch lassen sich Titel oder Überraschungseffekte wunderbar weich Ein- oder Ausblenden, wobei die „Fade-Time" stufenlos regelbar ist. Profis dürfte die Bandbreite der RGB-Wandlung von 5,5 Megahertz interessieren. An diese Käufergruppe hat man wahrscheinlich auch bei der Gehäusekonstruktion gedacht. Das 19-Zoll-Gehäuse eignet sich hervorragend für den Einbau in einem Videogeräteturm.

Davon einmal abgesehen gibt es keine weiteren Schwierigkeiten im Umgang mit dem VCG-3. Im Gegensatz zu anderen Genlocks kann man mit dem VCG-3 auch ohne Synchronimpuls eines TBC (Time-Base-Correctors) absturzfrei arbeiten. Somit ist es möglich, Computer-Bilder und -Animationen ohne Schnittplatz und TBC auf einem Videorekorder aufzuzeichnen.

(mms)

Info: Videocomp, Berger Straße 193, 6000 Frankfurt/M. 60, Telefon: 0 69/46 70 01

Assemblervergleich

Sieben Wege zum Prozessor

Aus einer Palette von sieben verschiedenen Tools kann man wählen, um die hohe Kunst der Maschinensprachenprogrammierung auszuüben. Stärken und Schwächen der einzelnen Verständigungshilfen zwischen Mensch und CPU soll folgender Vergleich aufdecken.

Verwirrung wird das Resultat sein, wenn der Maschinensprachenneuling nach dem richtigen Assemblerwerkzeug sucht. Mindestens sieben Programme bilden ein Puzzle mit allerdings unbekannten Formen und Farben. Um die Eigenschaften der einzelnen Programme zu entdecken, wurden sie einem harten Test unterworfen: im Einsatz bei einem großen Assemblerprojekt. Über eine Woche zog sich die Konstruktion des Sourcecodes hin, oft geändert, um für den nächsten Assembler in der Testreihe zu passen. Jeder Fehler, der dabei auftrat, wurde festgehalten. Denn Software-Werkzeuge, die Fehler enthalten, können einen Teil der geleisteten Arbeit vernichten oder, schlimmer noch, sich unterschwellig in dem fertigen Programm festsetzen. Überraschenderweise waren nicht alle Assembler mit den Standard-Include-Files von Commodore kompatibel (Diese Files enthalten die Konstanten und Strukturen, die der Amiga braucht).
Ohne diese Informationen ist es nicht möglich, Amiga-spezifische Programme zu schreiben.

MACRO Assembler 10.178 und 11.0

Der MACRO Assembler von Metacomco ist beachtenswert, weil er der erste und lange Zeit der einzige für den Amiga war. So wurde er zum Standard. Bis heute benutzt Commodore den MACRO Assembler, um neue Kickstart-Versionen zu entwikkeln; alle Include-Files sind speziell für den MACRO Assembler geschrieben. Da der MACRO Assembler ein wirklich gutes Produkt ist, löst er diese Aufgaben mit Bravour. Er hält sich genauestens an die Spezifikationen von Motorola, die den 68000-Chip entwickelt haben. Wichtige Erweiterungen wurden dazugefügt, um die Programmierung zu vereinfachen. Der MACRO Assembler ist jedoch langsam, er kriecht fast. Außerdem ging er nicht recht mit der Zeit: Version 10.178 wird mit Include Files für das ehrenwerte Kickstart 1.1 ausgeliefert. Metacomcos Version 11.0 ist mehr als doppelt so schnell wie die 10.178, aber strotzt vor Fehlern.

Die Dokumentation ist gespickt mit Ungereimtheiten und Lükken. Der ALINK-Linker ist ein komplexer und langwieriger Teil des Assemblierungsprozesses — es ist unmöglich, ohne ALINK ein lauffähiges Programm zu erstellen. Jedoch widmet sich das Handbuch auf keiner einzigen Seite diesem Programm. Eine andere Neuerung, der auf 18 Seiten ausführlich beschriebene Text-Editor, ist nicht beigepackt.
Eines der Beispielprogramme läuft nicht ohne größere Änderungen, die Include-Files sind seit Kickstart 1.1 nicht erneuert worden, und ein Fehler hält einen ohnehin davon ab, sie mit Version 11.0 zu assemblieren. Die nützliche Option „-e" arbeitet in dieser Version nicht, so ist es nur gut, daß diese nicht im Handbuch der Version 11.0 aufgeführt ist. Selbst wenn gegenteilige Befehle gegeben werden, hält die Expansion (das Auffädeln) der Macros an. Das bedeutet, daß sich ein Include-File nicht zweimal verwenden läßt. Werden zu viele Include-Files eingebunden, haucht der Stack ohne Warnung sein Leben aus und Fehlerflags wie „ASC.L (A0)" bleiben aus. Die Anweisungen, wie das System zu installieren und zu starten ist, sind unbrauchbar. Selbst wenn man bereits ein Metacomco-Experte ist, hat man nicht viel Glück mit dieser Version. Metacomco hat sich auf Fragen zu diesen Problemen nicht gemeldet. Briefe und elektronische Post blieben unbeantwortet.

DevPac AMIGA

DevPac ist ein sehr durchdachtes Softwarepaket. Es enthält einen integrierten Editor/Assembler, erlaubt aber auch den Betrieb unter CLI. Er kann Assemblercode direkt in ausführbaren Code übersetzen, und er kann auch linkbare Module oder absoluten Code (sinnvoll für das Brennen von ROMs und das Patchen bestehender Codes) erzeugen. Trotz der Tatsache, daß der DevPac von Sinclair QL und Atari ST portiert wurde, zeigt er nur wenig oder gar keine Fehler. Die Diskette enthält gute Beispiele für den Anfang, obwohl ein zusätzliches Beispielprogramm über die Behandlung von Daten im CHIP RAM willkommen wäre. Das Handbuch ist selbst für Anfänger verständlich. Der DevPac ist frei von Umständlichkeiten und unnützen Features. Die Fehlerbehandlung ist ein Genuß, so muß nur die rechte Amiga-Taste mit der J-Taste gedrückt werden, um zur nächsten Zeile mit einem Fehler zu springen. Der MonAm Debugger ist wie der Rest des Pakets klein, schnell und effizient. Er kann Debug Hunks lesen, den Guru verfolgen, Code einzeln abschreiten und Register und Speicher ausgeben. Obwohl Tracing (Zeitlupe) und Überwachungsfunktionen fehlen, erledigt MonAm seine Aufgabe gut. Es gibt nur zwei Klagen: Erstens können die Vorteile eines größeren Bildschirms nicht genutzt werden, weil die Screengröße fest ist, und zweitens stürzt die Suchfunktion beim A500 und A2000 ab, wenn sie das Gesuchte nicht findet.
Obwohl beim Assembler absolut keine Fehler gefunden wurden, unterliegt der DevPac doch einigen Einschränkungen. Diese sind alle klar und ehrlich im Handbuch aufgeführt. Zu den wichtigsten Einschränkungen gehört, daß keine lokalen Labels unterstützt werden, was ich als schmerzlich empfinde. Auch können die Labelnamen nicht länger als 16 Zeichen sein. Dies ist unverständlich, da einige ROM-Funktionen lange Namen haben; der DevPac kompensiert dies, indem er diese umbenennt. Auch ist nur eine SECTION-Direktive pro File erlaubt; dies ist aber akzeptabel, da ein Programm nötigenfalls in mehrere Files zerlegt werden kann. Zudem paßt der Editor seinen Arbeitsspeicher nicht dynamisch dem Text an, und es kann nur ein File bearbeitet werden.

CAPE 68K

Der APE 68K enthält einen integrierten Editor, der entweder als vereinfachte EMACs- oder Wordstar-Nachbildung arbeiten kann. Der Editor paßt den Puffer nicht dynamisch an, wenn Text eingefügt wird, ob-



Alle Kandidaten, hier noch für die Kamera posierend, unterwarfen sich der gleichen Härteprüfung.

wohl er die Fähigkeit hat, Files zu bearbeiten, die größer als der Puffer sind. Außer Pull-Down-Menüs werden keine Amiga-Funktionen genutzt, nicht einmal der Cursor kann mit der Maus bewegt werden. Sämtliche Einstellungen des Editors und des Assemblers können mit einem einzigen Kommando gespeichert werden. Mit der Ausnahme, daß nur ein File bearbeitet werden kann, ist der Editor leicht und bequem zu bedienen. Eine einzigartige Spezialität des CAPE 68K ist ein ARexx-Port. Besitzer des ARexx(William S. Hawes)programms können Editor und Assembler fernsteuern. Der Assembler ist nahezu hundertprozent Metacomco-kompatibel und geht sogar darüber hinaus. Dadurch wird es leicht, Programme, die für andere Assembler geschrieben wurden, zu konvertieren. Der CAPE verfügt über mehr Features als alle Mitstreiter. Eine besonders nette Option ist MACFILE, die wie INCLUDE arbeitet, aber erkennt, daß die meisten Include-Files nur einmal gelesen werden müssen (alle anderen lesen jedes Include-File zweimal). CAPE 68K unterstützt ein Paket von Optimierungen, die fast alles behandeln, mit der Ausnahme, daß long Branches nicht in short Branches umgewandelt werden können. Ebenfalls befindet sich der gleiche Assembler, CASM, ohne integrierten Editor auf Diskette. Zweimal stürzte der Editor während der Tests ab und zeigte auch einige kleinere Fehler. Der Assembler versäumt es, einen Fehler anzuzeigen, wenn eine Operation mulu.l ausgeführt werden soll. Die Option -o arbeitet nicht; dazu muß dann ein spezielles OBJFILE-Kommando in den SourceCode eingefügt werden. Der Hersteller Inovatic gibt eine Upgrade-Garantie, die es gestattet, die Originaldiskette gegen die jeweils neueste Version zu tauschen (eine neue Version kommt gerade auf den Markt).

Lattice ASM

Die Version 4.0 des Lattice-C-Compliers kommt in Verbindung mit einem Assembler auf den Markt. Es ist klar, daß dieser in erster Linie als Unterstützung für den Compiler gedacht ist und nicht als alleinstehendes System. Der ASM ist nur bedingt Metacomco-kompatibel, gerade genug, um mit den meisten Include-Files zu arbeiten, obwohl einige nicht korrekt compiliert werden. Macros werden nicht vollständig unterstützt. Es gibt keine temporären Variablen, Befehlsvarianten wie bchg.b sind nicht erlaubt, und ASM unterstützt so wichtige Assemblerdirektiven wie EQUR nicht. Die Konvertierung der Standards erfordert einen Haufen aufwendige Arbeit. Positiv ist, daß der ASM hervorragende Kontrolle über die Plazierung von Segmenten im CHIP oder FAST RAM gibt. Wie nicht anders zu erwarten, ergänzen sich die Eigenschaften des ASM mit denen des C-Compilers, so kann der Code Base-relativ oder für spätere Manipulationen mit BLINK erstellt werden. Der Lattice Assembler ist der einzige, der über einen Satz neuester, kommentierter Include-Files verfügt. Die Dokumentation jedoch ist schrecklich; auf ein paar dürftigen Seiten werden die wichtigsten Features nicht einmal gestreift. Auch die komplexe und kritische Direktive CSECT wird nicht behandelt; nirgends ist die Option -u erwähnt, die benötigt wird, um die Beispiele auf der Diskette zu assemblieren. Wenn bei dem Befehl MOVEM eine große Anzahl von Registern angegeben wird, enstehen Fehler in Code und Speicher. Wird beabsichtigt binäre Konstanten wie „Label equ %1011" zu verwenden, produziert ASM falschen Code. Der Befehl CNOP ist defekt und erzeugt fehlerhaften Code und ASM schmuggelt word- und long word-Daten auf ungerade Adressen.

Aztek AS

Der Aztek-C-Compiler erzeugt Code, indem er erst den Quelltext in Assemblerbefehle übersetzt und diese dann assembliert, so daß immer ein Assembler notwendig ist. In der aktuellen Version wurde er dahingehend verbessert, daß er nun eigenständig arbeiten kann. Die Kompatibilität zu Metacomco ist gut, Standard-Include-Files werden ohne Probleme behandelt, und er kann die meisten Sourcecodes ohne den geringsten Fehler assemblieren. Der Aztek bietet von allen Software-Paketen die beste Mög- ▶

lichkeit zur Code-Optimierung. Das Programm optimiert Vorwärts- und Rückwärtssprünge, Sprünge in Subroutinen und Verzweigungen genauso wie Transporte zu Datenregistern. Das einzige echte Problem ist das eigene Object-File-Format des Aztek, dies bindet den Benutzer an die Aztek-Programme. Der Linker des Aztek bietet die Möglichkeit, Standard-Object-Files des AMIGA einzubringen, der Assembler sollte eine Option zur Ausgabe solcher Files haben. Einige DCB.B-Befehle mit mehr als 160 Elementen erzeugen nicht korrekten Code. Es ist nicht möglich, mit MOVEQ Bereiche zu bearbeiten, MOVEQ #Ende-Anfang,d0 wird nicht ausgeführt. Selbst, wenn die Optimierung abgeschaltet ist, werden einige Code-Sequenzen trotzdem optimiert, das ist eine Sünde, die nur schwer zu vergeben ist.

ASSEM PRO

ASSEM PRO bietet eine einzigartige Möglichkeit: Wenn er während des Assemblierens einen Fehler entdeckt, erscheint ein Fenster, das die Eingabe der Korrektur erlaubt, und fährt fort. Dabei wird das File im Editor mitkorrigiert. Ab diesem Punkt jedoch beginnen die Probleme. So wurde die Portierung nicht sorgfältig genug betrieben, daß der nicht mit genügend Anstrengung auf den Amiga portiert wurde. ASSEM PRO kombiniert einen tief integrierten Editor, Assembler, Disassembler, Debugger und Quick-Reference-Utility in einem einzigen, riesigen Programm; eine starke Nähe zur ST-Version blieb erhalten. Wenn ASSEM PRO gestartet wird, öffnen alle vier Hauptmodule Fenster, die nicht einzeln geschlossen werden können. Der Editor ist peinlich langsam und fehlerhaft, so daß die Autoren des Handbuchs es für nötig hielten, auf mindestens einen Fehler hinzuweisen. ASSEM PRO stürzte 37mal ab, bevor ich aufhörte, weiter zu zählen. Wenn man spezielle Anforderungen vermeidet und Geduld mitbringt, ist es sogar möglich, etwas zu assemblieren. Der Assemblierungsvorgang läßt sich nicht automatisieren und CLI-Argumente werden nicht angenommen. Um ASSEM PRO verwenden zu können, muß der gesamte Programmkomplex geladen werden. Der Assembler ist weder zum Metacomco- noch zum wesentlich wichtigeren Motorola-Standard kompatibel. Es gelang nicht, auch nur ein einziges Source-File zu finden, das ohne Änderungen assembliert werden konnte. ASSEM PRO kann keine Standard-Include-Files lesen, ein umständlich erstelltes Custom Set befindet sich auf der Diskette. Diese sind allerdings veraltet, fehlerhaft und schlecht editiert. Der Debugger ist verhältnismäßig gut und hat eine vernünftige Aufmachung. Allerdings hat er Schwierigkeiten, sich in die Amiga-Umgebung einzupassen. Er kann keine Standard-Debug-Hunks lesen und hängt sich bei Multitaskingbetrieb auf. Diese Version von ASSEM PRO wird von allen sechs Mitbewerbern überflügelt.

(B. Nesbitt/T. Kestler)

Vergleich der sieben Assembler

	Metacomco 10.178	Metacomco 11.0	CAPE 1.0	AssemPro 1.0	Lattice 4.0	Aztek 3.6	DevPac 1.2
Base-relativer Code	N	N	J	N	J	J	N
Linkbarer Code	J	J	J	N	J	J	Y
Direkt ausführbar	N	N	J	J	N	N	Y
Absoluter Code	N	N	J	J	N	N	Y
68010 Prozessor	N	N	J	J	N	J	N
68881 Fließkomma	N	N	N	N	N	J	N
68020 Prozessor	N	N	N	N	N	J	N
Optimierung rückwärts	J	J	J	J	J	J	N
Optimierung vorwärts	N	N	N	N	N	J	N
Debugging-Symbole	J	J	J	N	J	J	Y
Debugger inbegriffen	N	N	N	J	N	J	Y
Geschwindigkeit(Sek.)*	60	16	16	33	41	19	10
Temporäre Labels	J	J	J	J	N	J	N
Flexibilität **	1	1	2	1	3	3	2
CHIP/FAST RAM	N	N	J	J	J	J	Y
Kompatibilität ***	100%	98%	99%	5%	60%	90%	85%
Dokumentation **	2	4	3	2	4	2	2
Speicherbedarf	103K	107K	50/140K	146K	99K	91K	21/33K

* Es wurde immer dasselbe Programm für alle Assembler benutzt.
** Noten: 1 = sehr gut 2 = gut 3 = befriedigend 4 = ausreichend
*** Zum Metacomco-Standard.

Vater tüftelt mit

Seit vier Monaten bin ich stolzer Besitzer eines Amiga 500. Zwei Computerkurse haben mich soweit gebracht, daß ich den größten Teil meiner Freizeit, wenn ich meine umfangreichen Schularbeiten fertig habe, (manche Lehrer spinnen, oder?) mit dem Amiga-Handbuch neben mir versuche herauszufinden, wie was funktioniert. Wenn ich mich mit meinen Freunden über Computer unterhalte, merke ich, daß ich mich schon ganz gut auskenne. Wie Ihr wißt, machen Spiele-Disketten ihre Runden oder man bekommt schon mal eine geschenkt. Eine hat mich ganz toll begeistert. Ich kann mit ihr eigene Spiele aufbauen und sie meinen Freunden vorführen. Mein Vater ist auch vom Computer-Virus befallen und so mancher Samstag wurde von uns beiden mit Probieren, Experimentieren, Eingeben, Löschen, Lachen, Staunen, Schimpfen, gegenseitigem Anstoßen und oft schlappem Aufgeben vernichtet. Vor einigen Tagen legte er mir die Amiga-Welt 1/89 auf mein Keyboard. Mit heller Begeisterung und voller Erwartung blätterte ich sie erst einmal durch und las auch ein paar Seiten, und dann wurde mir schwindelig.

M, Ihr macht ja eine schöne dicke und schlaue Zeitschrift für Leute, die schon 2 Computer zerhackt haben, aber kleinen Einsteigern ist kein leichter Beitrag gegönnt. Oder habe ich etwas übersehen? Naja, eine Autozeitschrift stellt auch keine Rollschuhe oder Fahrräder vor, „aber Kleinwagen für Führerscheinanfänger".

Also ließ ich alles noch Fachchinesische hinter mir und las ausführlich alle Artikel über Computerspiele durch. Ganz besonders interessierten mich vier Spiele: 1. Fairy-Tale, 2. Grand Monster Slam, 3. Mortville Manor, 4. Menace. Diese Spiele sind zwar für meinen Geldbeutel ganz schön löchrig, aber der nahe Geburtstag läßt mich hoffen.

Und nun komme ich zum eigentlichen Knackpunkt meines Briefes und möchte Sie um Ihre Hilfe bitten. „Schreiben Sie mir bitte, wo ich das eine oder andere Spiel beziehen kann und ob die vier genannten für meinen Computer geeignet sind."

C. Dieling,
Euskirchen

Speziell für Einsteiger ist unser BASIC-Kurs gedacht. In den nächsten Ausgaben werden wir weitere Einsteigerhilfen geben. Die genannten Spiele laufen alle auf dem Amiga 500. The Faiery Tale Adventure, Grand Monster Slam und Menace gibt es bei Rushware, Mortville Manor bei Bomico. Die Vertriebsadressen stehen auf der Kurz-notiert-Seite im Spieleteil.

Dritte und letzte Runde

Betr.: BOOT-ROM (1/88 und 3/88)

Lediglich der A1000 besitzt ein Boot-ROM, das im Speicherbereich ab $F80000 liegt und zu knapp 8 Kilobytes genutzt ist. Es lädt das Betriebssystem von der Kickstart-Diskette in den Speicherbereich von $FC0000 bis $FFFFFF und wird anschließend ausgeblendet. Es wird daher normalerweise NICHT von Anwendungsprogrammen angesprochen und ist natürlich auf einem A500 und A2000 NICHT vorhanden! Es gibt auch nur EIN trackdisk.device, EINE disk.resource, EINEN Wack etc. etc., und dies alles befindet sich in besagtem Kickstart-ROM, das KEINERLEI Ähnlichkeit — auch nicht teilweise — mit dem Boot-ROM des A1000 aufweist. Kickstart besitzt — in der gleichen Version — keine funktionellen oder strukturellen Unterschiede zwischen den bisherigen Rechnern der Amiga-Serie.

Die Antwort auf das ursprüngliche Problem („Warum befindet sich das trackdisk.device nicht im DEVS-Verzeichnis?") ist im Rahmen dieser Diskussion vielleicht etwas untergegangen: disk-residente Devices werden von ramlib.library durch die DOS-Funktion LoadSeg() geladen, die im Falle des internen Laufwerks wiederum auf Filing-System und trackdisk.device und Standard-Filing-System bereits im ROM befindlich sind und üblicherweise nicht nachgeladen werden können.

Betr.: Undokumentierte Systemroutinen (2/88)

Die Sammlung der FD-Dateien wurde mit Freigabe der endgültigen Entwicklerdisketten für Kickstart/Workbench 1.3 „aufgeräumt" und enthält nur noch die für diese Betriebssystem-Version relevanten Beschreibungen für Libraries und Devices.

Ein Mißverständnis resultierte aus der falschen Benennung der debug.lib: Die Endung „.library" ist shared Libraries vorbehalten, „.lib" kennzeichnet eine linked Library, die Teil des Load-Modules wird.

Wir können nur nochmals darauf hinweisen, daß zur Entwicklung ausschließlich die offiziellen Informationen herangezogen werden sollten, auch mit Blick auf die Verwendung der aktuellen Include-Files 1.3, notfalls 1.2, keinesfalls jedoch die Versionen 1.2.1! Ein strikt nach den Regeln von 1985 geschriebenes Programm läuft aller Wahrscheinlichkeit auch noch unter Kickstart 1.3, ganz im Gegensatz zu mancher High-Performance-Application (lies: Spiel/Kopierprogramm). Ausnahmen bestätigen die Regel.

Betr.: Addison-Wesley

Zum Zeitpunkt da dieser Brief geschrieben wird, ist das Erscheinen der überarbeiteten Ausgabe der Addison-Wesleys für Februar und März 1989 vorgesehen. Etwaige Irritationen unsererseits wären nur dann aufgetreten, wenn relevante Informationen ausnahmsweise einmal rechtzeitig erschienen wären, wichtig speziell für diejenigen, die auf offizielle Buchpublikationen angewiesen sind.

Der langen Rede kurzer Sinn: Viele Probleme und Mißverständnisse könnten vermieden werden, würden Zeitschriften und Magazine mehr Sorgfalt bei der Auswahl ihrer Leserbriefe und auch Artikel walten lassen, speziell bei Qualität und inhaltlicher Korrektheit. Auch die Verwendungen der üblichen Konventionen und das Lesen der Handbücher könnte hierzu beitragen.

D. Twigg-Flesner, Gronau
R. Babel, Taunusstein

Kein BASIC-Spaß

Es ist wirklich erstaunlich, mit welcher Unverfrorenheit den Lesern „Bücher" aufgeschwatzt werden, die sonst schon lange im Reißwolf gelandet wären. Man kommt unwillkürlich zur Vermutung, daß der betreffende Verlag ihnen (kleingeschrieben) einiges zukommen läßt, damit die Auflage nicht ganz im Sande sitzen bleibt.

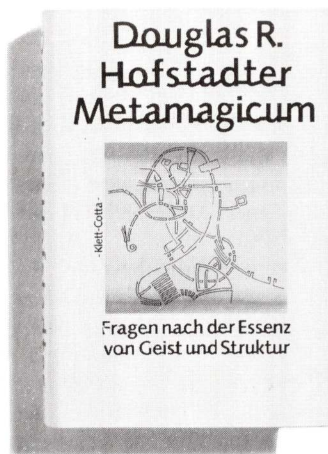
Hierbei beziehe ich mich speziell auf das Buch von David A. Lien: Programmierpraxis mit MS BASIC. Der Rezensent (R. Babel) hat das Buch nie im Leben gelesen, sondern seinen Text garantiert aus anderen Gazetten abgeschrieben, oder er lügt bewußt.

Wenn man liest: (Zitat) „Bei der Übersetzung ließ man es nicht an Sorgfalt fehlen..."), kriegt man fast einen Lachkrampf. Leider habe ich den horrenden Preis bezahlt, um damit BASIC zu lernen; die Lust ist mir aber schnell vergangen. Selten findet man ein Buch, das derart schlampig ausgeführt wurde. Zahllose Druckfehler, Zeilenvertauschungen, Auslassungen, usw., usw. machen es zur Qual, danach zu arbeiten. Viele Programme sind nicht lauffähig, weil falsch durch Syntaxfehler usw. Als Anfänger hockt man stundenlang, um so ein Programm zum Laufen zu bekommen. Da lob ich mir die Bücher von DATA BECKER oder vom M&T-Verlag — natürlich findet man auch da mal einen Fehler — Fazit: Das von Ihnen gelobte Buch ist für Einsteiger auf keinen Fall zu empfehlen.

Udo Teuchert,
Viernheim

Quarks und Änigmen

Douglas R. Hofstadter, Autor des Kultbuches „Gödel, Escher, Bach“, flicht das „endlos geflochtene Band“ weiter. Das neue Werk des Professors für Cognitive Science heißt „Metamagicum“. Seine „Fragen nach der Essenz von Geist und Struktur“ — so der Untertitel — sind eine Sammlung von 25 Artikeln. Sie sind in den Jahren zwischen 1981 bis 1983 im „Scientific American“ und in dessen deutscher Ausgabe, dem „Bild der Wissenschaft“, erschienen. Hofstadter ergänzte die Artikel durch Ausarbeitungen angrenzender Themen und durch 32 Postscripta. Diese Postscripta umfassen sowohl nachträgliche Erkenntnisse des Autors als auch Leserbriefe.



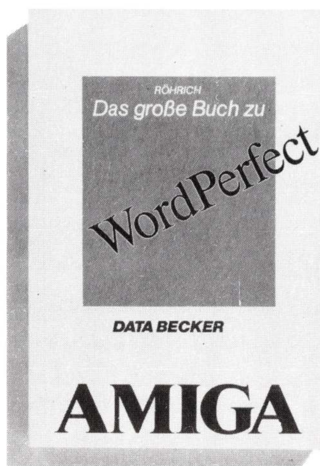
Auf 946 Seiten begleitet der Leser Hofstadters Suche nach der Grundstruktur des Universums, nach den zentralen Ordnungsprinzipien „da draußen“. Mit Hilfe der Mathematik werden inhärente Strukturen des Geistes und abstrakter Regelmäßigkeiten der Welt analysiert. Ausgehend von der These, der Geist sei das Haupterkennungsorgan für Strukturen und gleichzeitig Hauptproduzent bestimmter Arten von Strukturen, zergliedert Hofstadter die Welt. Wie der Autor wird nun auch der Leser vom Netz der Fragen nach der alles einenden Zauberformel gefangengenommen.

Und wer sich erst in den Schlingen und Schlaufen selbstbezüglicher Sätze verheddert hat, kann das Buch nicht mehr aus der Hand legen. Er liest weiter und begreift anhand der Bilder Eschers, was Aristoteles mit seinem Ausspruch „panta rei“ (alles fließt) meinte. In Eschers Metamorphose II werden Schriftzüge zu Schachfeldern, dann zu Lurchen, deren Umrisse sich zu Bienenwaben vergröbern. Die daraus schlüpfenden Bienen werden zu Faltern, deren Negativ-Konturen Fische darstellen. Und alles entwickelt sich zurück zum Schachmuster, das wiederum zu den anfänglichen Schachfeldern reduziert wird. Dann läßt der Leser sich von Rubiks Zauberwürfel und seinen Lösungsalgorithmen verzaubern. Darauf versinkt er in der Theorie des Chaos, wirbelt in Turbulenzen und läßt sich von seltsamen Attraktoren anziehen. Er lernt, daß LISP eine klare und intelligente KI-Programmiersprache ist, die den Weg zum klareren Denken ebnet. Anhand der Heisenbergschen Unschärferelation wird das Denken weiter geschärft. Experimente am Turing-Änigma lassen Zweifel an der künstlichen Intelligenz aufkommen. Während man von den Ereignissen in der amerikanischen Kleinstadt Heppington liest, lernt man die Gesetze der Wahrscheinlichkeit kennen. Dann wird diskutiert; Achilles und die Schildkröte ereifern sich über die Relativität ihrer Existenz. Und zum Schluß fragt Hofstadter nach dem ICH.
Gute Fragen sind der Weg zu richtigen Antworten. Wer das „Metamagicum“ liest, lernt vielschichtig zu fragen. Vieles bleibt jedoch unbeantwortet. Der Leser wird feststellen, daß auch dies eine Antwort ist — eher zumindest als die wackeligen Gedankengebäude, die Gewißheit vorgaukeln.
(ub)

Metamagicum, Douglas R. Hofstadter, Klett-Cotta, Stuttgart 1988, 946 Seiten, 56 Mark, ISBN 3-608-93089-2.

WordPerfect wird durchsichtig

Der aktuelle Trend zum Gestaltungssystem, dem sogenannten Publishing, macht auch vor traditionellen Textprogrammen nicht Halt. WordPerfect bietet auf dem Amiga so viele Funktionen, daß dieses System mit der Abwicklung alltäglicher Korrespondenz unterbelastet ist. Die wahren Stärken des Programms werden erst im Bereich anspruchsvoller Dokumentationen deutlich. Hier fällt angenehm auf, wie leicht sich Grafikdateien in Texte einbinden lassen.

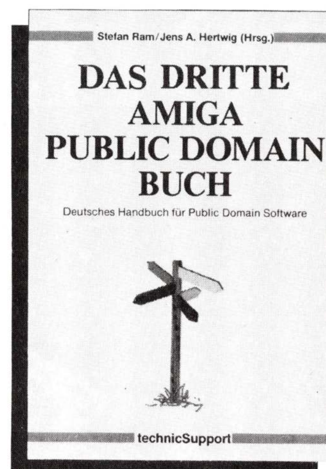


Kompliziert und gewöhnungsbedürftig ist die Wahl der Funktionen. Sie werden durch eine Mischung aus Tastenkombinationen und verschachtelten Menüs aktiviert. Soviel Komplexität verlangt nach tiefgehender Dokumentation — und die bietet Peter Röhrich mit seinem Buch „WordPerfect“. Er beginnt mit einfachsten Funktionen. Im ersten Abschnitt wird Editieren, Speichern und Drucken von Texten erklärt. Der zweite Abschnitt befaßt sich mit allen Funktionen von WordPerfect, die der weiteren Bearbeitung dienen. In den Anhängen finden sich Funktionsübersichten, mit deren Hilfe festgestellt werden kann, wie eine bestimmte Funktion aktiviert wird.

WordPerfect, Peter Röhrich, Data Becker GmbH, Düsseldorf, 1988, 316 Seiten, 39 Mark.

PD-Hilfe, Numero III

Band I und Band II des Großen Public Domain Handbuches erklärten mehr als 180 frei kopierbare Programme. Und der Bedarf an Anleitungen wächst weiter. Pünktlich zur CeBIT ’89 wird technicSupport sein drittes PD-Buch herausbringen und darin über 90 neue Public-Domain-Programme dokumentieren. Software aller Anwendungsbereiche wurde ausgesucht und leicht verständlich erklärt. Unter den ausgewählten Programmen befinden sich Dateiverwaltungen, Mal- und Grafikprogramme, Texteditoren, Terminkalender, Biorhythmus, Modelleisenbahn-Steuerung, Assembler, C-Compiler, fraktale Grafik, CLI-Hilfen und Utilities, Spiele und witzige Screen-Hacks.



Am Ende des Buches sind alle erhältlichen PD-Disketten aufgelistet und kommentiert. Die Liste wurde aktualisiert und enthält alle FISH-Disks bis 172, RPD-Disks bis 150, ACS-Disks bis 86, Panorama-Disks bis 30c und FAUG-Disks bis 50. Außerdem werden Update-Informationen zu den ersten beiden PD-Büchern geboten.
Für 95 Mark sind auf zehn Disketten die im Buch beschriebenen Programme erhältlich.

Das dritte Amiga Public-Domain-Buch, R. Leithaus/J. Hertwig (Hrsg.), technicSupport, Berlin 1989, 416 Seiten, 49 Mark, ISBN 3-926847-06-9.

Animationen der High-Tech-Klasse

Animationen, wie sie bisher nur auf Workstations möglich waren, lassen sich mit Caligari realisieren. Die zum Test vorliegende Vorversion bewies, daß der vorgesehene Verkaufspreis von 2000 Dollar gerechtfertigt erscheint.

Drahtmodelle lassen sich von allen Seiten und aus allen Perspektiven betrachten, wie auch an Angelpunkten bewegen.

Caligari von Octree soll neue Maßstäbe im Bereich der Animation setzen. In drei Jahren Entwicklungsarbeit entstand ein hochkomplexes Animationsprogramm, das auf die Ansprüche von Profis zugeschnitten ist, das jedoch so bedienerfreundlich konzipiert ist, daß selbst Kinder damit arbeiten können.

Dank einer neuen Eingabemethode, bei der die Ansicht von drei Seiten (wie bei Sculpt oder Turbosilver) überflüssig wird, können Objekte bewegt und im Blickwinkel des Betrachters in Echtzeit verändert werden. Die momentane Demoversion enthält bereits alle Funktionen der vorgesehenen Endversion — außer Speichern und Laden. Daß die endgültige Version 2000 Dollar kosten soll, wird dafür sorgen, daß Caligari in professionellen, bestenfalls in semiprofessionellen Videoanimationsstudios zum Einsatz kommt. In diesem Bereich ist der gehobene Preis des Produktes tragbar, sogar recht günstig. Caligaris zentrale Einheit ist das Eingabemodul. Damit werden Objekte geladen, plaziert und kombiniert. Ist die Szene fertig gesetzt, kann die eigentliche Animation beziehungsweise deren Planung erfolgen. Der letzte Arbeitsschritt bei jeder Animation ist die Aufzeichnung auf Video. Optional kann dazu Caligaris integrierte Script-Sprache verwendet werden. Zur Aufzeichnung der Animation wird eine serienmäßige Schnittstelle geboten, die mit verschiedenen Videorecordern und Genlocks zusammenarbeitet. Falls der Videorecorder mit einem sogenannten Frame-Buffer ausgestattet ist und somit die Aufnahme von Einzelbildern erlaubt, können die Animationen vollautomatisch berechnet und auf Band überspielt werden.

Caligari ist zur Zeit nur in der NTSC-Version verfügbar. Ein für europäische Bedürfnisse unhaltbarer Zustand, der die Arbeit an einer PAL-Version forcieren dürfte.

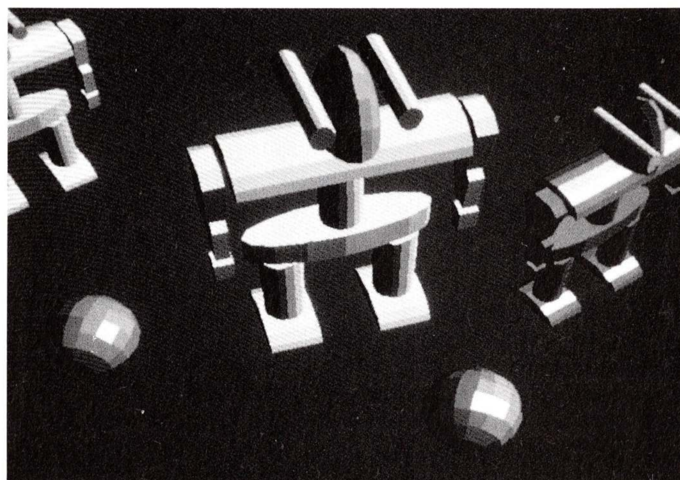
Objekte, die mit Caligari bearbeitet werden sollen, können entweder aus einer Objektbibliothek stammen oder auf einem zweidimensionalen Feld selbst entworfen werden. Es ist aber auch möglich, Objekte aus mehreren Primär-Objekten wie Kreisen, Würfeln und Pyramiden aufzubauen. Diese Objekte können in allen Dimensionen vergrößert und verkleinert werden, so daß sich vielfältige Einsatzmöglichkeiten ergeben.

Objektgröße und Perspektiven sind frei wählbar

Normalerweise arbeitet man mit Caligari auf einer dreidimensionalen Ebene, deren dritte Ebene schräg im Bildschirm liegt. Auf dieser Ebene kann ein Objekt mit der Maus in allen drei Richtungen bewegt werden — für eine Bewegung in der dritten Dimension muß zusätzlich die linke Maustaste gedrückt werden. Die perspektivisch korrekte Größe des Objekts verändert sich dabei in Echtzeit entsprechend der Lage auf der Ebene. Es kann auch eine zweidimensionale Ansicht aus drei Ansichten aufgerufen werden.

Alle anderen Funktionen werden bei Caligari über Menüs ausgeführt. Es gibt insgesamt ein knappes Dutzend Menüs, die alle eine Besonderheit aufweisen. Der Benutzer kann nämlich die Untermenüs, die er öfters braucht, auf dem Bildschirm belassen. So kann jeder seine eigene Benutzeroberfläche zusammenstellen, die zum Beispiel das Hauptmenü und die zwei aktuellen Untermenüs gleichzeitig darstellt.

Objekte können in Caligari genauso hierarchisch angeordnet werden wie in Sculpt-Animate oder ähnlichen Programmen. So können sich bei der Bewegung eines Armes alle Finger mitbewegen. Bewegt man jedoch nur einen Finger der Figur, bleibt der Arm ruhig. Das ist die Voraussetzung für eine realitätsbezogene Animation.
Jederzeit kann die Szene aus dem aktuellen Blickwinkel, der auf drei verschiedene Arten mit der Maus im Raum verändert



Erst für die Aufzeichnung auf Video werden die Flächen mit Farbe aufgefüllt.

wird, unter Berücksichtigung aller Farben und Lichtquellen erstellt werden. Caligari ist jedoch kein Raytracing-Programm, sondern berechnet lediglich aktuelle Farben ganzer Flächen, so daß Effekte auftreten, die denen von Videoscape1.0 ähnlich sind. Ist das Bild einmal berechnet — was auch ohne mathematischen Coprozessor sehr schnell geht — und fügt man der Szene keine Objekte zu, ergibt sich eine wesentlich geringere Rechenzeit für den nächsten Blickwinkel. Tests haben ergeben, daß eine Szene mit zirka 1000 Polygonen auf einem Standard-Amiga in 30 Sekunden berechnet wird. Diese Zeit wird sich jedoch mit dem von Octree angekündigten Raytracing-Modul erheblich erhöhen. Dann sind auch mit Caligari echte Raytracing-Animationen möglich. Den Objekten können nun auch bestimmte Eigenschaften wie Farbe oder glänzendes Material zugewiesen werden.

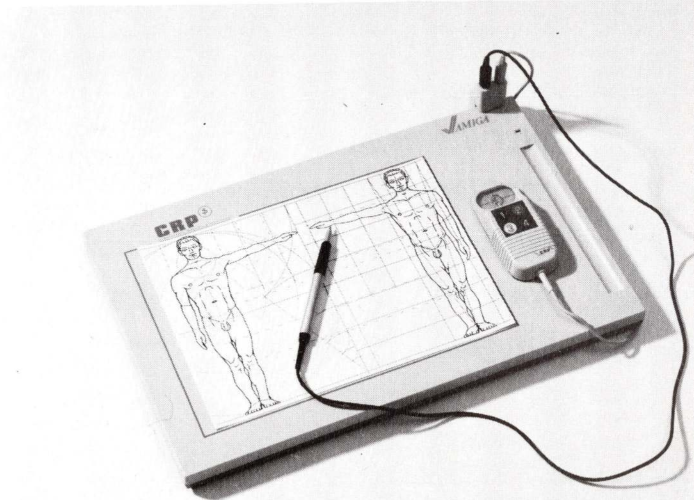
Hobbyversion wird von Electronic Arts vertrieben werden

Caligari hält selbst in seiner jetzigen Version — die vom Hersteller als prerelease verkauft wird — viele Überraschungen bereit. Mit den für die nächsten zwölf Monate angekündigten Erweiterungen wie Raytracing-Modul und das Script-Modul wird Caligari neue Animations-Standards für den gesamten Computerbereich setzen.
Interessant für Heimanwender und Amateure: Eine abgemagerte Version von Caligari soll demnächst in den Vertrieb von Electronic Arts gehen. Über die Preisvorstellungen ist noch nichts bekannt.
(Ottmar Röhrig)

Digitaler Zeichenblock

Wer den typischen Schwung aus dem Handgelenk mit Stift und Pinsel oder auch maßstabsgetreue Übertragungen in der Bitmap wiederfinden möchte, benötigt statt der Maus ein Grafiktableau.

Paintprogramme, die lediglich Maussteuerung bieten, eignen sich nicht für Präzisionsarbeiten. Flächige Malereien werden durch die etwas vage Linienführung mit der Maus nicht beeinträchtigt. Wer aber versucht, detaillierte Abbildungen wie Konstruktionspläne, Logos und Landkarten von der Vorlage abzuzeichnen oder Federzeichnungen zu imitieren, wird mit der unpräzise arbeitenden Maus nicht zurechtkommen. Für diesen Zwecke werden jetzt Grafiktabletts angeboten. Eines davon ist der CRP-Digitizer. Der besteht aus CRP-Zeichenbrett, Metallstift, Fadenkreuz-Cursor und der dazugehörenden Treiber-Software.
Das CRP-Zeichentablett hat für den Computerzeichner die Funktion eines Blattes Papier. Mit dem — einem Kugelschreiber ähnlichen — Metallstift wird direkt auf dem Tablett beziehungsweise auf einer darauf befestigten Vorlage gezeichnet.



Ein Grafiktablett wie das CRP ist für präzise Grafikarbeiten unentbehrlich.

Mit Pauspapier oder Fadenkreuz

Jede gezogene Linie erscheint sofort am Monitor. Exakt ausgedrückt, handelt es sich bei dem Digitizer um ein Koordinatenerfassungsgerät, das Punkte in binärer Form an den Rechner weiterleitet und in Bildschirmdaten umwandelt.
Wem das Abzeichnen im Stil des Abpausens mit Kohlepapier zu ungenau ist, der benutzt den Fadenkreuz-Cursor, um Vorlagen parallaxefrei abzutasten.
Der Einsatz des Digitizers ist jedoch nicht auf den reproduktiven Bereich beschränkt. Seine besonderen Stärken liegen in der kreativ-künstlerischen Anwendung. Durch die ergonomischen Vorzüge des Stiftes gegenüber der Maus läßt sich der individuelle Zeichenstil viel besser darstellen. Besonders deutlich wird diese Fähigkeit bei der Eingabe von Handschriften. Erscheint die Schrift bei Mausführung krakelig und uneinheitlich, so lassen sich mit dem Stift schwungvolle und sogar schnörkelige Schriftzüge mit persönlichen Charakteristika eingeben. Daran hätten selbst Graphologen ihre Freude.
Trotz aller Vorzüge — ein Grafiktablett ist kein Blatt Papier. Man sollte sich also nicht mit dem Ellenbogen auf die Grafikfläche stützen, da das Programm aussetzen könnte. Weiterhin ist zu bedenken, daß ein Digitizer nur eine gewisse Geschwindigkeit nachvollziehen kann. Zu schnell darf also nicht geschrieben oder skizziert werden. Wer sich daran gewöhnt hat, wird schnell neue Einsatzgebiete für Malprogramme entdecken.
Das CRP-Grafiktablett mit Treiber kostet im Format DIN-A4 950 Mark, im Format DIN-A3 1649 Mark. Das Anschlußkabel kostet 55 Mark, der Metallstift 185 Mark, das Fadenkreuz mit vier Funktionen 185 Mark, mit acht Funktionen 295 Mark. Zu allen Preisen sind 14 Prozent Mehrwertsteuer hinzuzurechnen.

(mms)

Info: Cybernetical Research & Production, Konstanz, Tel. 0 75 31/6 33 96.

Lattice C Compiler nun in der Version 5.0

Nicht ganz ein Jahr nach dem Erscheinen der Version 4.0 von Lattice C für den Amiga erschien jetzt die neue Version 5.0 dieses populären Compilers, der nicht zuletzt auch für die Programmierung des Betriebssystems des Amigas benutzt wurde.

Gegenüber den alten Versionen des Compilers hat sich in der Aufmachung und im Umfang des Entwicklungssystems einiges geändert. Der Compiler und der Linker wurden überarbeitet, alte Fehler beseitigt (und neue eingebaut). Alle Werkzeuge des Compiler Companion und der Lattice Screen Editor werden jetzt mitgeliefert. Außerdem steht jetzt allen Lattice-Usern ein neues unschätzbar wertvolles Werkzeug für ihre Arbeit zur Verfügung, der neue Source Level Debugger „CodePRobe", mit dem sich C-Programme im Quelltext Tracen lassen; kein Lattice-Fan muß jetzt noch neidisch auf den Manx Aztec C Compiler schielen.
Einige der wichtigsten Neuerungen sollen etwas genauer beleuchtet werden. Das Handbuch von Lattice C wird jetzt nicht mehr mit Spiralrücken ausgeliefert, sondern kommt in zwei Ringbücher geteilt. Es werden jetzt fünf Disketten ausgeliefert, auf denen sich das komplette System einschließlich aller Werkzeuge, aller Includefiles und aller Libraries befindet, zuzüglich verschiedener Beispielprogramme. Die erste Diskette ist bootfähig und die weiteren Disketten sind so eingerichtet, daß sie mit drei Diskettenlaufwerken arbeitsfähig sind. Minimal läßt sich das System zwar mit einem Diskettenlaufwerk und 512 KByte Speicher benutzen, spätestens wenn man Source Level Debugger benutzen will, sollte man über mehr Speicher verfügen. Optimal ist eine Festplatte und mehr als zwei MByte RAM-Speicher.

Der Compiler 1c (1c1, 1c2) und der Global Optimizer go
Die Entwickler Lattice trieb ihre Anstrengungen, den Richtlinien des ANSI-C-Standards zu folgen, weiter voran. Die beiden Schlüsselworte „const" und „volatile" werden unterstützt. „const" beschreibt einen Datentyp, der konstant ist, das heißt nach der Initialisierung nicht mehr vom Programm verändert werden darf. „volatile" bezeichnet Daten, die von außerhalb des Programmes verändert werden können, und daher vom Compiler nicht zur Codeoptimierung in einem Prozessorregister gehalten werden dürfen, sondern bei jedem Zugriff aus dem Speicher geholt werden müssen. Sonst könnte es passieren, daß sich das Programm einen Wert in einem Register behält und diesen für Berechnungen verwendet, obwohl der eigentlich gültige Wert im Speicher schon längst verändert worden ist.
Prototypen von Funktionen dürfen jetzt auch optional den Namen von Parametern enthalten, und eine variable Anzahl von Argumenten kann durch eine Ellipsis (. . .) dargestellt werden.
Interessant vor allem für Programmierer, die hauptsächlich für den AMIGA entwickeln, und nicht auch für andere Computersysteme verwendbare (portierbare) Programme schreiben müssen, sind die neuen Schlüsselworte. Mit dem Schlüsselwort „chip" wird der Compiler angewiesen, die Daten in das von den Customchips adressierbare Chip-RAM zu laden. „near" und „far", die sich auf die Adressierung von Datenobjekten beziehen. „near" erlaubt dem Compiler explizit ein Datum 16-Bit-Adreßregister-relativ zu adressieren, oder in Verbindung mit Funktionsaufrufen Programmzähler-relative Unterprogrammaufrufe. „far" bewirkt sozusagen das Gegenteil und legt fest, daß alle Zugriffe auf ein Datenobjekt über 32-Bit-Adressen erfolgen müssen.
Mit „regargs" wird der Compiler angewiesen Funktionsaufrufe mit Parameterübergabe in Registern auszuführen. „stdargs" bewirkt die normale Übergabe von Parametern über den Stack. „SAVEDS" definiert ein Unterprogramm, das das Adreßregister A4 beim Aufruf mit dem korrekten Wert laden muß. Das ist zum Beispiel beim Starten von neuen Tasks mit AddTask( ) notwendig.
Die Fehlermeldungen des Compilers wurden in ein Textfile ausgelagert, um die Anpassung des Systems an die jeweilige Landessprache zu erleichtern. Einige der deutschen Meldungen sind allerdings etwas kurios. Wer weiß, was folgende Meldungen bedeuten sollen: 'Markennamen Konflikt' (Schleichwerbung?), 'Konvertierung eines riesigen Zeigers zu einem weit entfernten Zeiger' (Zeiger in die Wüste geschickt?) , 'Variable nicht riesig, konvertiert in die Ferne'.
Alle Prozessoren der Motorola 68k-Serie und die Mathe-Coprozessoren werden von dem Compiler unterstützt. Um den neuen Source Level Debugger zu unterstützen, kann der Compiler verschiedene Mengen an Debugginginformationen erzeugen (vollständige Informationen -d3), die die Länge und Laufzeit des Programmes mehr oder weniger vergrößern. Der Programmierer kann damit selbst genau entscheiden welche Debuggingaufgaben er mit CodePRobe lösen möchte.

Auch Abstürze können optimiert werden.

Mit der -g-Option wird der Compiler veranlaßt ein Cross-Referenz Listing auszugeben, -O startet den Global Optimizer, der versucht das Programm kürzer und schneller zu machen, während der Debugging und Testphase ist diese Option gefährlich, da der Optimierer ein „funktionsfähiges" Programm auch zu abstürzenden Schrott wegoptimieren kann, und das Lesen des disassemblierten Codes wesentlich erschwert. Mit -prp wird der Compiler veranlaßt für alle in einem Quelltext vorhandenen Funktionen eine Datei mit den Prototypen zu erstellen, anstatt das gesamte Programm zu compilieren.
Gegenüber der Version 4.0 ist jetzt die Berechnung von

Adressen einzelner Elemente eines Arrays geändert worden, das größer als 32 KByte ist, da der Compiler nun versucht, den Offset sechzehn Bit Basisrelativ zu ermitteln, was bei großen Arrays immer schief gehen muß.

Der Assembler asm
Der Assembler unterstützt jetzt alle Prozessoren der 68k-Familie einschließlich der Mathe-Coprozessoren und des 68030. Werden irrtümliche 68020 Opcodes im 68000 Modus verwendet, so meldet man eine Warnung, bei 68020 Opcodes „vergißt" asm jedoch die Warnungen auszugeben.

Der Linker blink
Der Linker unterstützt mit dem Schlüsselwort ADSYM den Source Level Debugger CodePRobe, und kann jetzt mit Hilfe des neuen Startup Moduls cres.o reentrante Programme unterstützen, das heißt Programme, die man mit dem CLI-Befehl „resident" resident in den Speicher laden kann.
Beim Erzeugen von reentrantem Code tut sich blink allerdings im Moment noch schwer. Linkt man seine Programme mit dem Modul cres.o in Verbindung zum Beispiel mit der Option Batch, so fängt der Linker an zu linken und linkt, und linkt, und linkt . . . , bis die RAM-Disk, Disketten oder die 40-MByte-Festplatte mit seiner Linkdatei voll ist; Control-C funktioniert natürlich auch nicht mehr. Das geht bei der RAM-Disk recht schnell und ist relativ ungefährlich. Sollten Sie aber auf der Diskette oder Festplatte linken, und in Ihrer Verzweiflung den Rechner neu booten, so haben Sie unausweichlich einen Schreibfehler auf ihrer Diskette oder Harddisk. Diskdoktor bei einer 40-MByte-Harddisk funktioniert übrigens normalerweise auch nicht mehr (zu wenig RAM-Speicher im AMIGA).
Mit Hilfe des Startup-Moduls catch.o und des Utilities tb (TraceBack) ist es manchmal möglich, einem fehlerhaften Programm noch einen Post-mortem-Dump zu entreißen, bevor es abstürzt. Das ist eine Datei, die, mit Hilfe von TraceBack gelesen, Informationen über die Fehlerbedingungen geben kann, die zum Ausstieg führten.

Der Source Level Debugger cpr
CodePRobe erlaubt, den Ablauf eines C- oder Assemblerprogramms im Quelltext mitzuverfolgen, auch die gemischte Darstellung des C-Quelltextes und des daraus erzeugten Assemblerprogramms ist möglich. Der Debugger öffnet wahlweise einen eigenen Screen oder benutzt bei Speicherplatzmangel die Workbench für seine Ausgabefenster. Beim Start des Debuggers erscheint je ein Fenster für den Quelltext des Programms und ein Fenster für den Dialog mit dem Benutzer. Optional kann man zwei weitere Fenster öffnen, eines dient zur Anzeige des Inhalts der Prozessorregister, und im anderen kann man sich den Inhalt verschiedener Variablen laufend anzeigen lassen.
Der Programmierer kann das Programm entweder Instruktion für Instruktion abarbeiten lassen, wobei er beliebig zwischen C-Quelltext und Maschinensprache hin- und herschalten kann, oder sogenannte Watchpoints setzen, mit denen er die Veränderung des Inhalts von Variablen überprüfen kann. Ferner ist es möglich, den Inhalt des Programm- und Datenspeichers zu untersuchen, und gegebenenfalls Änderungen am Inhalt von Variablen vorzunehmen. Außerdem kann jederzeit der Typ einer Variablen mit dem Befehl „whatis" abgefragt werden (zum Beispiel: whatis w 4 Byte Pointer to a 404208 Byte Structure World).
Während des Testlaufs eines Programms übernimmt der Debugger volle Kontrolle über das Programm. Während das Programm läuft, schaltet der Debugger das Fenster oder den Screen des Programms nach vorne, damit der Programmierer das Verhalten des Programms ansehen kann; Erreicht das Programm einen Breakpoint, eine Variable, einen bestimmten Wert oder nach dem Abarbeiten einer Programm- oder Assemblerzeile im Einzelschrittmodus, holt CodePRobe den eigenen Schirm wieder nach vorn, und wartet auf neue Eingaben. Über ein Interface zur Macrosprache ARexx läßt sich der Debugger auch extern steuern.
Bei der Ausgabe von Variablenwerten wird im Gegensatz zum Source Level Debugger von Manx Aztec C nicht nur der Inhalt von Strukturen und Aufzählungstypen (enum) korrekt ausgegeben, sondern auch Arrays vollständig angezeigt, was der Aztec-Debugger nicht beherrscht. Das einzige Problem dabei ist, das man diese Ausgabe nicht anhalten kann. Falls man also den Inhalt einer 404 208 Bytes großen Struktur ausgeben läßt, kann man sich derweil getrost pensionieren lassen . . .
Weiterhin werden jetzt folgende Werkzeuge mit dem Compiler mitgeliefert:
— cxreff, ein Utility zum Erstellen vom Cross-Reference Listings für C Quelltexte.
— diff zum Feststellen von Unterschieden zwischen Texten.
— extract zum Ausgeben von Filenamen innerhalb eines bestimmten Directories.
— fd2pragma, schon von 4.0 bekannt. Erzeugt Prototypen für die AMIGA Betriebssystemfunktionen aus den Dateien, die auf der Extras Diskette als Zubehör zu AMIGA-Basic ausgeliefert werden, um mit Hilfe der „≠pragma"-Anweisung des Compilers die Betriebssystemfunktionen direkt ohne die amiga.lib aufrufen zu können.
— files, zum umfassenden Manipulieren von Dateien (List, Rename, Delete).
— grep, zum Suchen und Ausgeben von sogenannten Regular Expressions (Textmustern) in Textdateien.
— lmk, Makeutility um das Compilieren und Linken großer Programmprojekte zu managen.
— lse, der Editor.
— lstat und lprof, Werkzeuge um das Laufzeitverhalten von Programmen auf deren Effizienz zu überprüfen. Man kann durch die erstellte Statistik erkennen, wieviel Zeit das Programm in den einzelnen Unterprogrammen verbringt.
— oml, Objekt Module Librarian.
— omd, Objekt Module Disassembler.
Das neue Entwicklungssystem Lattice C 5.0 gibt dem Softwareentwickler auf dem AMIGA ein komplettes, leistungsfähiges Werkzeug in die Hand, das weitgehend den ANSI-Standard unterstützt, und darüber hinaus für Programmprojekte speziell auf dem Amiga sehr geeignet ist, da es mit vielen auf dem AMIGA angepaßten Erweiterungen aufwartet. Auch ein Neuling sollte sich von der erschlagenden Fülle der Möglichkeiten beim Einstieg in die Sprache C mit diesem Entwicklungssystem nicht abhalten lassen, denn mit Hilfe des wohlstrukturierten englischsprachigen Compilerhandbuches sollte auch er die für den Anfang wichtigen Informationen finden, um mit dem C Compiler umzugehen. Das Handbuch kann und soll auch hier kein Lehrbuch über die Sprache C ersetzen.
Ein Vergleich mit dem Manx Aztex C Compiler V3.6 läßt zwar auf den ersten Blick im Umfang fast keinen Unterschied erkennen, arbeitet man aber längere Zeit mit dem Manx Compiler, findet man seine Schwächen gegenüber Lattice C 5.0. Der Aztec C Compiler erkennt wesentlich weniger strukturelle Schwächen und Fehler von C-Programmen als der Lattice Compiler, die dieser mit vielen Warnungen montiert. Der früher in Zeitschriften oftmals zu unrecht festgestellte Geschwindigkeitsvorteil von Aztec C-Programmen ist höchstens marginal, durch die wesentlich größere Erfahrung der Firma Lattice im Compilerbau für die Sprache C und der Codeoptimierung ist das ausführbare Programm des Lattice C Compilers im Gegenteil meist kompakter und schneller als der von Aztek erzeugte Code.

(Wilfried Häring)

Man spricht Amiga

Vielfältig sind die Möglichkeiten, mit dem Prozessor zu kommunizieren. Die Wahl der richtigen Fremdsprache fällt da nicht leicht. Im babylonischen Gewirr der Amiga-Hochsprachen hilft dieser Leitfaden bei der Entscheidung.

Manche Programmiersprache, die für den Amiga zu haben ist, wurde ursprünglich für andere Computer entwickelt. Zum Beispiel Pulic-Domain-Sprachen wie Meta oder Small C, die auf jedem Computer laufen sollen und den Unterschied zwischen Intuition und (göttlicher) Eingabe nicht kennen. Nicht alle Sprachen beherrschen daher Amigas Fähigkeiten. Drei grundlegende Fragen erleichtern den Vergleich der angebotenen Programmierdialekte:

1. Wie gut erlaubt die Sprache den Zugriff auf die Amiga-Rom-Kernel-Funktionen (die Routinen, die Fenster öffnen und Menüs erscheinen lassen? Eine Sprache, die einen angemessenen Zugriff auf die System-Calls bietet, sollte fähig sein, jede Art von Software auf dem Amiga zu realisieren.
2. Ist die Sprache oder deren vorliegende Version benutzerfreundlich? Die Sprache sollte flexibel und schnell sein.
3. Wie einfach ist die Sprache zu erlernen? Wie sieht die Lernkurve aus? Bei der Arbeit mit mehreren Sprachen ist es wichtig, sich nicht erst jahrelang mit den Besonderheiten vertraut machen zu müssen.

Fragen, ob Compiler oder Interpreter, die Codegröße und die Ausführungsgeschwindigkeit sind nicht die wichtigsten Kriterien. Absicht dieser Gegenüberstellung ist es nicht, in den Krieg der Entwickler einzugreifen. Statt nach Geschwindigkeitsdaten und Komprimierungsrekorden werden die Sprachen nach Eigenschaften wie Bequemlichkeit, Umgebung und Informativität beurteilt. Es werden hier nur die Sprachen diskutiert, die deutlich die Spezialfunktionen des Amiga unterstützen. Dabei sind nicht alle Sprachen erfaßt, die diesem Kriterium gerecht werden.

Die Sprache C

C ist das ursprüngliche Amiga-Entwicklungswerkzeug. Viele Teile des Amiga-Betriebssystems sind in C geschrieben. Obwohl Modula 2 nun zur Herausforderung antritt, ist C noch immer die populärste Sprache unter den Entwicklern für Amiga-Software. Jede Art der Manipulation läßt sich mit C verwirklichen. Der erzeugte Code ist schnell genug, um viele zeitkritische Probleme zu lösen. Der Programmierstil ist jedoch nicht leicht zu verstehen. Das Layout des Programmtextes ist völlig frei, und die Namen der Funktionen sind mehr intuitiv. Viele C-Programmierer benutzen kurze, fast kryptografische Variablennamen, so daß erst eine Vielzahl von Kommentaren Überblick verschafft. Ein alter Witz geht um, daß C-Programmierer selbst Namen wie „Ort“ noch zu „Ot“ verkürzen. Es ist nicht einfach, C zu erlernen, dafür aber höchst lohnenswert. Es gibt zwei große Implementierungen von C für den Amiga: den Lattice- und den Manx-(Aztek-)C-Compiler. Beide enthalten die nötigen Dateien, um den Amiga zu programmieren, bieten aber keine Lernprogramme. Literatur und Ratschläge von Freunden müssen herangezogen werden, bis C beherrschbar wird.

Der Manx-C-Compiler

C-Ängste werden von dem Manx-Package noch verstärkt. Das Handbuch enthält alle Informationen, ist daher aber sehr umfangreich und kompliziert. Bereits das Einfügen von einfachen Trennblättern, würde dem Novizen beim andauernden Nachblättern helfen. Der Manx-Compiler erlaubt, den Code in vier verschiedenen Speichermodellen zu schreiben. So lassen sich die Größen der Code- und Datenbereiche mit einem hohen Grad an Flexibilität kontrollieren, was jedoch manchen eher irritiert. Der Manx ist wirklich ein effizienter C-Compiler. Das Handbuch enthält einige wirklich gute Beispielprogramme und sogar ein Kapitel über den Programmierstil in C. Die Programmierumgebung ist sehr gut an das CLI angepaßt. Eine typische Compilation würde so aussehen:

1. Laden des favorisierten Texteditors und des C-Source-Programms.
2. Editieren der Files und Abspeichern der Änderungen. Falls Sie weniger als ein MB RAM haben, verlassen Sie den Editor.
3. Starten Sie den Manx-C-Compiler, und geben sie diesem ▶

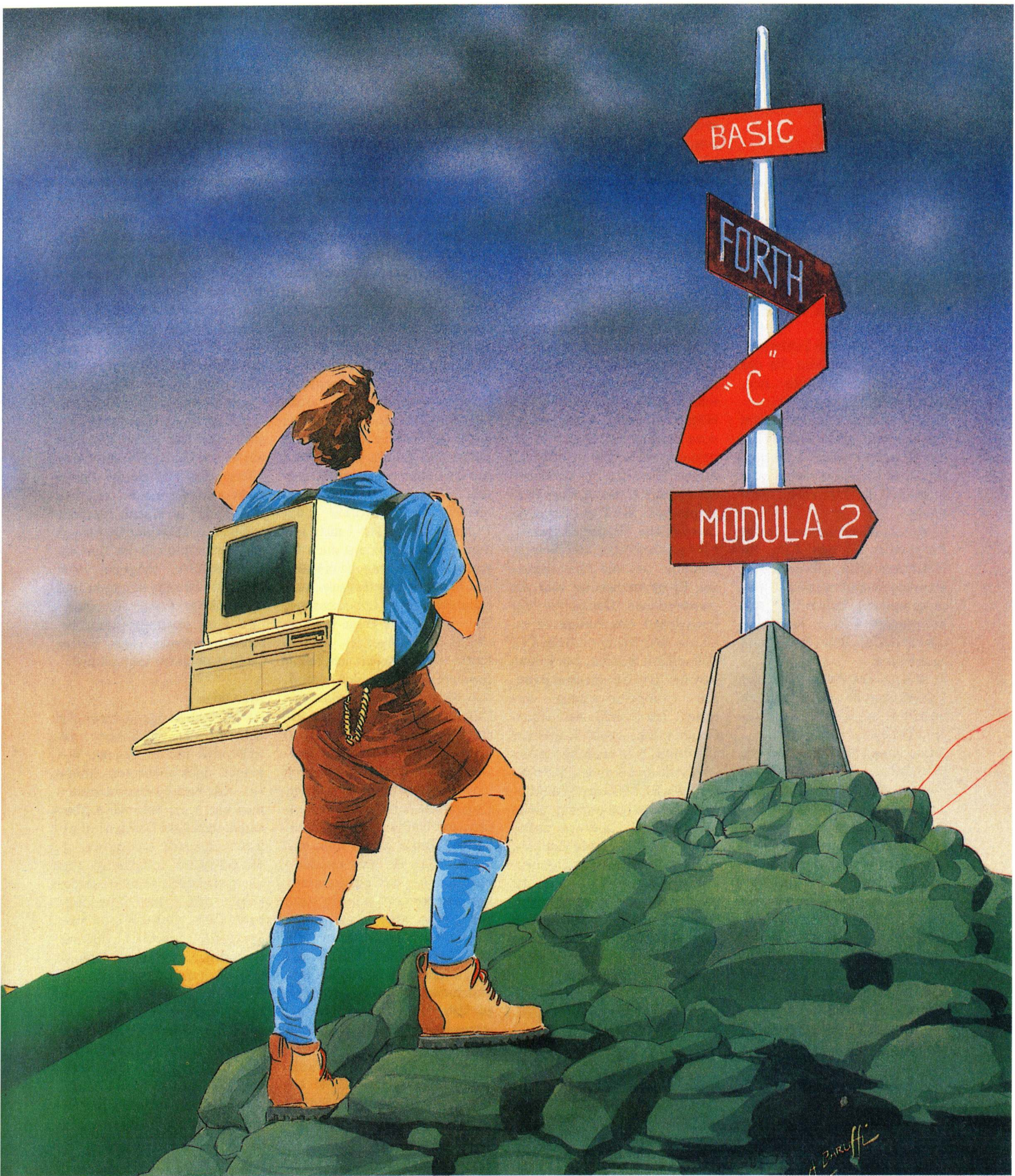
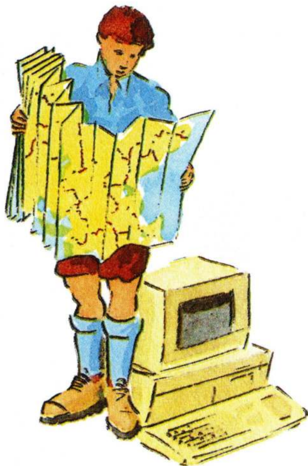


Illustration: Andrea Baruffi



den Namen des Files an, das sie compilieren wollen.
4. Beachten Sie alle Fehlermeldungen, und editieren Sie das Programm nötigenfalls neu, falls erforderlich.
5. Der Manx-Compiler startet automatisch und assembliert das Ergebnis vom Compiler (Ihr C-Source-Code wird vom Compiler in Assembler übersetzt).
6. Abschließend linken Sie Ihr Programm mit den Aztec-Bibliotheken (eine Bibliothek ist hier eine Sammlung von Unterprogrammen) indem Sie den beigefügten Linker benutzen, um ein ausführbares Programm zu erhalten.
7. Starten des Programms zur Fehlerkontrolle. Gehen Sie zu Schritt 1. zurück, bis das Programm perfekt ist.

Aztec unterstützt nahezu alle Fähigkeiten des Amiga. Sie können ROM-Kernel-Funktionen aufrufen, indem Sie einfach den Namen der Funktion und die benötigten Parameter eingeben. Während des Linkvorgangs werden die „Klebe"-Routinen aus der Aztec-Bibliothek gelesen (Eine „Klebe"-Routine ist eine kleine Routine, die sicherstellt, daß alle 68000-Register richtig vorbereitet werden, bevor der Systemaufruf erfolgt). Zum Beispiel übergibt C alle Parameter einer Funktion auf den Hardware-Stack, aber die Systemroutinen benötigen die Parameter aus den CPU-Registern. Die „Klebe"-Routinen bewegen die Parameter vom Hardware-Stack in die CPU-Register und vergeben die Kontrolle an die aufgerufene Funktion, dadurch wird die aufrufende Software mit den Systemroutinen „verklebt". Eine sehr gebräuchliche Komponente des Aztec-Systems ist der Source-Level-Debugger. Debugging heißt „Entwanzen", es ist das Suchen und Beseitigen von Fehlern. Mit dem Source-Lèvel-Debugger müssen Sie nur den Source Code überprüfen, um die Bugs (Fehler) zu finden.

Lattice C

Lattice C, das originale Amiga-C, das von Commodore an alle Entwickler vertrieben wurde, hat eine Reihe von Revisionen hinter sich. Die Dokumentation führt gemächlicher in die Sprache C ein als der Manx-Compiler. Die ersten drei Seiten beschreiben, wie man technische Hilfe von Lattice erhält. Insgesamt ist das Handbuch angenehmer, aber nicht ausführlich genug. Die neueren Versionen des Lattice C lassen dem Programmierer die Wahl der Speichermodelle. Es gibt einige sauber geschriebene Beispielprogramme, um die Organisaton der Programmierung und die verschiedenen Kategorien der Funktionen des Amiga zu erklären. Wiederum muß das CLI aufgeräumt werden, bevor man sich ans Arbeiten machen kann. Ein Compiliervorgang entspricht im Prinzip der Manx-Compilation, etwas langsamer allerdings. Wie auch Manx bietet Lattice eine gute Unterstützung der ROM-Kernel-Funktionen über eingebundene „Klebe"-Routinen, aber es unterstützt auch die Möglichkeit, ROM-Routinen direkt aufzurufen, ohne den Weg über die „Klebe"-Routinen zu nehmen. Dies macht die Ausführung etwas schneller (Zur neuesten Version von Lattice C siehe Bericht in diesem Heft).

Modula 2

Modula 2 wurde von Niklaus Wirth als Nachfolge für sein populäres Pascal entwickelt. Mit Modula 2 wird das Programm in Module zerlegt, von denen jedes eine spezielle Funktion hat. Ein Modul besteht aus zwei Teilen: ein Definitionsteil, der angibt, welche Variablen das Modul braucht und zurückgibt und das Programm-Modul, das die Anweisungen enthält, die die Aufgabe ausführen. So lange der Definitionsteil nicht verändert wird, kann der Programmteil beliebig geändert werden, und der Rest der Software harmoniert mit dem Modul. Modula 2 zwingt dazu, ein Projekt als Ausdruck von Modulen zu analysieren und nimmt es auch sehr genau, damit Parameter richtig an Funktionen übergeben werden. Das kann zunächst ziemlich frustrierend sein, da die Sprache sehr pedantisch scheint. Mit etwas Erfahrung in der Programmierung in Moula 2 werden Sie den größten Vorteil entdecken. Wenn ein Programm erfolgreich compiliert und gelinkt wurde, wird es fast immer einwandfrei laufen. Modula-2-Compiler vermeiden 90 Prozent der Programmierfehler — nicht nur Syntaxfehler, sondern auch logische Fehler. Manche empfinden Modula 2 als langsam, umständlich und hypersensibel, aber das ist eine Frage des persönlichen Geschmacks. Andere Kritik — wahrscheinlich besser fundiert — stößt sich daran, daß diese Sprache initialisierte Datenstrukturen nicht unterstützt, beispielsweise Sprachen wie C, die direkt Teile des Speichers für Daten zugänglich machen. In Modula 2 muß das Programm Ausdrücke verwenden, um Daten im Speicher zu halten wenn das Programm startet. Dies ist nicht unerheblich, da sich das Programmieren des Amiga sehr oft stark auf solche Strukturen stützt. Betrachten wir die drei wichtigsten Compiler auf dem Markt: TDI Modula 2, M2Amiga und Benchmark Modula 2. Jeder ist ein komplettes System zur Softwareentwicklung, und alle benutzen „Klebe"- Routinen zur Anpassung an die ROM-Funktionen, wie vorher für C beschrieben. Hier enden jedoch auch schon die Gemeinsamkeiten.

TDI Modula 2+

TDI veröffentlichte den ersten Modula-2-Compiler für den Amiga. Es begeisterte viele, die etwas Besseres als BASIC und etwas Leichteres als C suchten. Dafür ist es ein extrem unergonomisches Entwicklungssystem. Von den 350 Seiten des Handbuchs befassen sich weniger als 40 Seiten mit der Programmierung in Modula 2 und dem System. Der Rest besteht aus dem Beispielprogramm eines rotierenden Würfels und den Listings der Definitionsmodule aller Amiga-spezifischen Funktionen und Include-Files. Wer Modula 2 mit diesem System lernen möchte, wird um weiterführende Literatur nicht herumkommen. Der Compiliervorgang ähnelt den C-Prozeduren. Eine wichtige Verbesserung weist die Fehlersuche auf: Der TDI-Editor liest aus einem Fehler-File des Compilers, der bei einem Fehler nicht schlapp macht, sondern alle Fehler in einem Atemzug auflistetet. Die Einarbeitung in die Gesamtmaterie ist nur mit Expertenhilfe möglich. Der Compiler ist sehr langsam, und der Speicher, den es für temporäre Files benötigt, kann beachtlich werden. Der fertige Code läuft dafür sehr schnell.

M2Amiga

Bei der Anfertigung einer Arbeitsdiskette beginnen bei M2Amiga die Probleme. Wer jedoch nach mehreren Anläufen seine Arbeitsversion zusammen hat, wird angenehm überrascht sein. Das Demo ist in wenigen Minuten compiliert, da die Software vollständig über die Workbench genutzt werden kann, falls dies gewünscht wird. Dazu wird das Icon eines Sourcefiles angeklickt und zur Bearbeitung geladen. Danach wird nach dem Abspeichern der Änderungen das Symbol des M2Amiga-Compilers mit Doppelklick aktiviert, um den Compiler zu starten, wobei dann beliebig viele Module nacheinander compiliert werden können. Schließlich wird durch Anklikken der Object-Files dem Linker mitgeteilt, welche Object-Files angebunden werden sollen. Dies macht die Arbeit mit M2 für Anfänger sehr leicht.

Das System kann auch vollständig unter CLI ohne Iconerzeugung laufen. Die Dokumentation erklärt gut die Benutzung des Editiors, der eine vereinfachte Version des EMACs darstellt. Wie der TDI-Compiler, kann auch M2 ein Fehler-File lesen und ermöglicht das Abschreiten der einzelnen Probleme. Das Handbuch zeigt, wie der Compiler selbst die Programme organisiert und geht etwas auf die Grundlagen von Modula 2 ein. Die Vertriebsfirma, Interface Technologies, bietet zusätzliche Software wie etwa einen Source-Level-Debugger, Spezialbibliotheken und ein Entwicklerpacket. Das System M2 scheint sich gut für die Entwicklung von Software zu eignen.

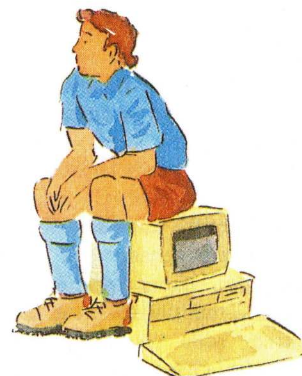
Benchmark Modula 2
Der dritte Modula-2-Compiler ist für den Geldbeutel der beste. Das integrierte Paket von Benchmark enthält Editor, Compiler und Linker. Am Linker findet man kaum etwas zum bekritteln. Der Editor basiert ebenfalls auf dem EMACs, aber er wurde ausgiebig verbessert. Nach dem Editieren wird über eine Funktionstaste das Programm in den Compiler geladen und sehr flott compiliert. Wenn Fehler gefunden werden, kann schrittweise durch das Programm gegangen werden, um die Ursachen zu finden. Eine zweite Taste startet den Linkvorgang und eine dritte führt das Programm aus — alles, ohne den Editor verlassen zu müssen. Einige der Editor-Kommandonamen sind etwas schwer zu behalten, aber es ist immer Hilfe über die Help-Taste oder das Handbuch erhältlich. Das Handbuch von Benchmark, über 500 Seiten stark, beschäftigt sich sehr ausführlich mit der Benutzung des Editors und des gesamten Systems. Es enthält ebenfalls Listings der Definitonsmodule der Amiga-Funktionen. Die gesamte Umgebung ist derart angenehm und leicht zu benutzen, daß Einarbeitungs- und Entwicklungszeiten stark vermindert werden. Es ist denkbar, daß ein Anfänger mit diesem System nach ein paar Stunden eigene Programme schreiben kann. Der Schöpfer von Benchmark, Avant Garde Software, bietet auch eine Reihe zusätzlicher Software an, die das Programmieren einfacher macht, beispielsweise eine vereinfachte Intuition-Bibliothek, eine Bibliothek mit C-ähnlichen Funktionen, um die Zeit der Übersetzung von C in Modula 2 zu verringern, und eine Bibliothek mit IFF- Routinen.

BASIC

BASIC hat einen schlechten Ruf. Die Sprache sei sehr lokker, nicht standardisiert, gut genug für für Leute, die nicht das Zeug zu haben. Vielleicht war einmal etwas Wahres daran, aber diese Ansicht muß revidiert werden. Jede BASIC-Version für den Amiga weist einige Neuigkeiten und Verbesserungen auf. Nahezu die gesamte Power des Amiga läßt sich ausschöpfen. In bezug auf die Ausführungsgeschwindigkeit erzeugen mindestens zwei Versionen einen Code, der wirklich vernünftig läuft. Sie sind sehr leicht zu erlernen und umgänglich. Die einzige Kritik, die man an BASIC finden kann, ist, daß es zu einem schlechten Programmierstil verleitet, was aber nicht in allen Fällen zutrifft. Lesen Sie deshalb erst einmal weiter, bevor Sie daran gehen, Ihre Extras-Disk neu zu formatieren, um Platz für angeblich sinnvollere Programmierwerkzeuge zu schaffen.

True BASIC
Ein strukturiertes BASIC benutzt moderne Programmierideen und Konstrukte. True BASIC hält sich selbst an eine modularisierte und einfache Form. Es benutzt einen dezenten Editor mit einfachen Cut- und Paste-, Such- und Ersatz-Funktionen. Vom Editor aus kann das Programm gestartet, verfolgt oder compiliert werden. Die Compilierungsoption speichert den erzeugten Code in den Buffer des Editors und erlaubt das Abspeichern auf Diskette. True BASIC compiliert Programme nicht in Maschinen-Code, sondern in einen computerunabhängigen Pseudo-Code. Das heißt, wenn mit True BASIC geschriebene Software vertrieben werden soll, muß eine Lizenz für den Runtime-Code von True BASIC beantragt werden. Die Dokumentation für True BASIC ist sehr professionell. Durch die Verwendung von großen Buchstaben in zwei ringgebundenen Teilen ist es leicht zu lesen, gut geordnet und informativ. Wenn die Software den Handbüchern entsprechen würde, wäre es ein gutes BASIC. Leider habe ich herausgefunden, daß der Compiler sehr langsam ist, andauernd fehlerhaft, Zufallsfehler bei großen Files produziert und einmal sogar den Textbuffer angriff, in dem mein Source-File stand. Die Sprache selbst unterstützt einen Befehlswortschatz, der nach Angaben der Hersteller völlig kompatibel zu True BASIC auf dem IBM-PC, Atari-ST und MacIntosh ist. Es ist auch ein Entwicklerpacket erhältlich, daß den Zugriff auf alle Amiga Spezialfunktionen über sogenannte „Klebe"-Routinen bietet, natürlich wird dadurch der Code unportabel. Dies bringt mich auf meine Hauptklage über dieses BASIC: Der portable Befehlswortschatz reicht nicht aus, um irgendetwas zu programmieren, daß den Ansprüchen des Amiga gerecht wird. Die Line-, Box- und Draw-Typen sind einfach zu allgemein und durch Benutzung der teueren Entwickler-Routinen geht die Portabilität verloren. True BASIC ist akzeptabel, obleich AmigaBASIC die gleichen Low-Level-Zugriffe auf den Amiga bietet. Wenn Sie aber Programme schreiben wollen, die aussehen und laufen wie Programme auf den vier, zur Zeit von True BASIC unterstützten Maschinen, dann wäre dies Ihre Traumsprache. Aber ist das wirklich die Art von Software, die auf dem Amiga sinnvoll ist? True BASIC Inc. bietet ebenfalls eine Reihe nützlicher Bibliotheken an, für Aufgaben der Stringmanipulation, 3D-Grafik sowie für das Sortieren und Suchen.

AmigaBASIC
Das BASIC, das mit dem Amiga ausgeliefert wird, mag eine gute Sprache sein, gut genug für alles, was Sie tun möchten. Gut dokumentiert (eine enorme Fehlermeldungsliste inclusive), ist die Amiga-Version des Microsoft-BASIC eine zuverlässige, strukturierte Ausführung, die auf eingebaute Amiga-spezifische Funktionen zugreift. BASIC ist eine Interpretersprache, aber mit AC/BASIC V 1.3 lassen sich selbstständige, lauffähige Programme erstellen. Ich hasse jedoch den Editor, der dem des MacIntosh nachgeahmt wurde. Der Cursor ist zu klein und die Maussteuerung so umständlich, daß es kaum möglich ist, eine bestimmte Position im Programmtext zu bestimmen. Am besten benutzt man einen anderen Editor und nimmt den AmigaBASIC-Editor nur zur Beseitigung von Syntaxfehlern. Zusätzlich zu den Amiga-spezifischen Befehlen hat AmigaBASIC über die FD-Files die Möglichkeit, auf fast alle ROM-Kernel-Funktionen zuzugreifen. FD heißt „Function Definition", die FD-Files werden von Commodore vertrieben, eines für jede ROM-Library. Es gibt verschiedene Files für Intuition, Exec, Graphics und so weiter. Diese Files enthalten alle notwendigen Funktionen und die Zugriffe auf die CPU-Register. Amiga-BASIC benötigt diese Information in Form der BMAP-Files, so daß Commodore ein Programm zur Konvertierung mitliefert. Nach der Konvertierung können ROM-Kernel-Funktionen direkt vom BASIC aus angesprochen werden, in dem die entsprechende Library geöffnet ▶



wird. AmigaBASIC ist leicht erlernbar und ermöglicht ein einfaches Debugging mit der Trace-Funktion. Wenn Sie ein gute Software entwickelt haben, lohnt es sich, sie mit einem Compiler zu bearbeiten.

AC/BASIC Compiler V 1.3

Die neueste Version dieses Compilers läßt BASIC-Programm so schnell ablaufen, wie man es noch nicht gesehen hat. Dieser Compiler übersetzt ein Programm in 68000-Maschinencode in kurzer Zeit. Da kein spezieller Editor besteht, muß mit dem ED- oder dem BASIC-Editor gearbeitet werden. Um Amiga-BASIC-Software mit dem Compiler verwenden zu können, müssen einige Änderungen der Syntax und unter Umständen der Programmierung erfolgen. Wenn jedoch Code für den Compiler geschrieben wird, wird dieser fast immer perfekt mit dem Interpreter arbeiten.

Ich habe in Version 1.2 eine Vielzahl solcher Diskrepanzen zwischen den beiden BASICs gefunden, aber in V 1.3 scheint dies korrigiert zu sein. Allerdings führt AC/BASIC auch neue Befehle und Kontrollstrukturen ein, so daß hier neue Inkompatibilitäten entstehen können. Normalerweise müssen die AC/BASIC-Runtime-Files immer vorhanden sein, damit die compilierte Software richtig läuft. Jedoch kann so compiliert werden, daß der Runtime-Code automatisch hinzugefügt wird. Dies schafft vollkommen unabhängige, lauffähige Programme, allerdings muß eine kostenlose Lizenz von Absoft eingeholt werden, um Files mit eingebautem Runtime-Code zu vertreiben. Das exzellente Handbuch erklärt den Compiler und alle Befehle sehr sorgfältig. Es ist voll von guten Beispielen und guten Ideen. Der Compiler läuft sicher und ist mausgesteuert sehr leicht zu bedienen. AC/BASIC erledigt den Zugriff auf die ROM-Funktionen genauso wie AmigaBASIC, daher gibt es hier keine Kompatibilitätsprobleme. Durch volle Systemkontrolle und die Geschwindigkeit der Maschinensprache kann man so in BASIC programmieren, daß der Eindruck einer „richtigen" Programmiersprache entsteht.

FBASIC

Völlig neue Wege geht der BASIC-Compiler von Delphi Noetic Systems Inc. Er produziert aus einem BASIC-Source-File den schnellsten Code und bietet sowohl eine Vielzahl von Programmier- und Optimierhilfen als auch einige Funktionen zu einer Amiga-gerechteren Programmierung. Diese Sprache benötigt umfangreiches technisches Verständnis und erfordert wegen seiner komplexen Fähigkeiten und der CLI-Schnittstelle lange Einarbeitungszeit. 70 anschauliche Beispiele erleichtern den Einstieg. Einige der Neuerungen von FBASIC sind in anderen Dialekten nicht zu finden beispielsweise Rekurdion, Records (C-ähnliche, auf dem Amiga sehr nützliche Strukturen), Pointer, Include-Files und direkter Zugriff auf die 68000-Register. Diese müssen nicht unbedingt genutzt werden, aber sie sind da, falls sie gebraucht werden. Ebenso Funktionen, um einfach mit Fenstern, Screens und Menüs zu arbeiten, aber dazu muß erst erlernt werden, wie Teststrings im Speicher organisiert sind. Auch diese Funktionen sind begrenzt, so daß der Blick auf die Nutzung der ROM-Routinen durch FBASIC fällt. Dies kann durch Einfügen der einfachen Anweisung „&SYSLIB01" in das Source-File geschehen, um danach mit einem FBASIC-Befehl die gewünschte Library zu öffnen. Wenn die richtigen Libraries geöffnet sind, kann jede ROM-Kernel-Funktion aufgerufen und die angegebenen Parameter übergeben werden. Es ist notwendig, die Runtime-Library immer sowohl beim Compilieren als auch beim Starten compilierter Programme bereitzuhalten (Der Compiler selbst ist in FBASIC geschrieben!). Wenn ein FBASIC-Code mit Runtime-Code vertrieben werden soll, muß eine Lizenz von zehn Dollar pro Produkt erworben werden. Die Hersteller von FBASIC verrieten, daß eine bessere Version des Compilers, ein Source-Level-Debugger und eine „FBASIC Logic Programming Extension" in Vorbereitung sind.

Wer ein komfortables BASIC sucht, ist mit der Kombination AmigaBASIC und AC/BASIC-Compiler gut bedient. Wenn

Bezugsquellen

C

Lattice C Compiler 4.0
Lattice
2500 S. Highland Ave.,
Suite 300
Lombard, IL 60148
001/312/916-1600

Aztec C68K/am-p
Manx Software Systems
One Industrial Way West
Eatontown, NJ 07724

Source Level Debugger
Manx Software Systems

Modula-2

Benchmark Modula-2
Avant Garde Software
2213 Woodburn
Plano, TX 75075
001/214/964-0260

M2Amiga
A+L AG
Im Späten 23
CH-8906 Bonstetten/ZH
0041/1/7003037

TDI Modula-2
TDI Software Inc.
10355 Brockwood Rd.
Dallas, TX 75238
001/214/340-4942

BASIC

A/C Basic Compiler
Absoft
2781 Bond St.
Auburn Hills, MI 48057
001/313/853-0050

True BASIC
True BASIC Inc.
39 South Main St.
Hanover, NH 03755
001/603/643-3882

F-Basic
Delphi Noetic Systems Inc.
PO Box 7722
Rapid City, SD 57709-7722
001/605/348-0791

Forth

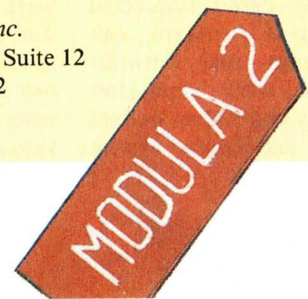
Multi-Forth
Creative Solutions Inc.
4701 Randolph Rd., Suite 12
Rockville, MD 20852
001/301/984-0262

JForth
Delta Research
PO Box 1051
San Rafael, CA 94915
001/415/485-6867

Prolog

Prolog Philgerma
Barerstr. 32
8000 München 2
089/28 12 28

Deutscher Vertrieb
der amerikanischen Produkte
über Computerfachhandel
(siehe Anzeigen)



mehr Power und Geschwindigkeit gewünscht werden, ist FBASIC die richtige Wahl, sofern man genügend Zeit zur Einarbeitung hat.

Pascal

Pascal wurde von Niklaus Wirth völlig neu entwickelt, ohne auf bestehende Sprachen Rücksicht zu nehmen. Daher konnte Pascal absolut nach einer vorher gefertigten Grammatik entwickelt werden. Dies führte dazu, daß Pascal für alle Anweisungen das gleiche Format besitzt und eine enorm saubere Struktur aufweist. Wichtig war eine Trennung der Bereiche Definition und Ausführung, die dann in Modula 2 noch weitergeführt wurde. In Pascal müssen zu Beginn eines Unterprogramms die dafür nötigen Variablen deklariert werden, Konstanten müssen am Anfang des Programms stehen. Die Unterprogramme haben zwei Gesichter: Die „procedure“ die vorwiegend ausführenden Charakter hat und die „function“, die als Routine zur Erledigung bestimmter Probleme (Berechnung ...) dient. Betriebssystemaufrufe können in Pascal leicht als procedures oder functions realisiert werden. Pascal ist leichter zu erlernen als C und auch schneller zu beherrschen. Bisher teilen sich zwei Pascal-Compiler den Markt: Der MCC-Pascal-Compiler und UCSD Pascal.

MCC-Pascal
Der MCC-Compiler besteht aus der Programmdiskette und dem englischen Handbuch. In Kürze wird er mit deutschem Handbuch erscheinen. Die Diskette selbst ist nicht startfähig, und es ist empfehlenswert, eine Arbeitskopie mit dem CLI-Befehl „install“ zu bearbeiten, um sie booten zu können. Dabei sollte dann ein Pfad mittels der Startup-Sequence festgelegt werden. Mitgeliefert wird nur der DOS-Editor ED 1.14, der für größere Programme unzureichend ist. Hier ist es sinnvoll, auf PD-Programme wie den MicroEMACs-Editor auszuweichen. Das englische Handbuch ist übersichtlich und geordnet, wenn es auch umfangreicher sein könnte (300 Seiten). Der Compiler wird mit dem ALINK-Linker geliefert, BLINK wäre etwas moderner. Mit 134K ist der Compiler nicht gerade klein und auch nicht der schnellste, aber er compiliert zuverlässig und unterstützt eine Fehlerausgabe. MCC Pascal bietet einen einwandfreien Zugriff auf die ROM-Kernel-Funktionen, die sich sehr leicht programmieren lassen. Dazu ist allerdings das „Amiga Reference Manual: Libraries and Devices“ nötig. Es bestehen für alle Bibliotheken Include-Files, die auf Pascal zugeschnitten wurden. MCC Pascal ist ein Compiler, der dank der leichten Erlernbarkeit von Pascal und der guten Anpassung an den Amiga vor allem Einsteigern das Programmieren auf dem Amiga erleichtert. Der Hersteller täte allerdings gut daran, den Compiler auf zwei Disketten mit einem guten Editor zu liefern, von denen eine startfähig ist.

Prolog

Prolog wurde entwickelt, um logische Konstrukte und Verknüpfungen besser programmieren zu können. Die Pioniere Clocksin und Mellish beschrieben eine Sprache, die für die Entwicklung künstlicher Intelligenz wie geschaffen war. Der grundlegende Unterschied zu anderen Sprachen ist die vollständige Rekursion von Prolog. Dadurch werden Problemlösungen möglich, die sich in anderen Sprachen nicht realisieren lassen. Nachteil solcher Rekursionen ist der hohe Speicherbedarf, so daß Prolog viel Speicher benötigt. Dafür können jedoch Fakten, Klauseln und Regeln bestimmt werden, die aufwendige Konstrukte sparen helfen. Prolog wird auf anderen Rechnern zur Entwicklung von Expertensystemen eingesetzt, auf dem Amiga ist es noch ein Neuling. Ich bin der Ansicht, daß Prolog dem Amiga helfen könnte, neue Aufgabengebiete zu ergründen.

Philgerma PROLOG
Im April '88 wurde der Philgerma-PROLOG-Interpreter veröffentlicht, ohne daß davon richtig Notiz genommen wurde. Grund: Prolog ist eine noch relativ unbekannte Sprache, und gerade auf dem Amiga zählte vor allem C. Doch auf einem Rechner wie dem Amiga könnten Programme mit künstlicher Intelligenz (KI) interessant sein. Der günstige Anschaffungspreis des Interpreters ist ebenfalls ein wichtiges Argument in der Frage nach der geeigneten Sprache. Da zur Zeit die Version 1.0 vorliegt, ist klar, daß noch Kinderkrankheiten auszustehen sind. Das größte Manko ist das zu knapp geratene Handbüchlein. Auch läuft der Interpreter nur mit 1 MB RAM! Die ROM-Kernel-Funktionen werden nur gering unterstützt. Als Editor dient auch hier der DOS-Ed. Die Oberfläche des Interpreters ist nicht gerade professionell, obwohl ein Debugger hier Pluspunkte schafft. Eine Sprache wie Prolog fordert in letzter Instanz professionelle Anwendungen, so daß ein Compiler all denen zur Verfügung stehen sollte, die ihre Programme vertreiben wollen. Der Interpreter bliebe dann ein günstiger Einstieg. Solange jedoch das Handbuch nicht drastisch erweitert ist, bleibt Prolog nur etwas für Freaks. Gut gefiel mir, daß eine PD-Diskette mit verschiedenen EMACs-Versionen und Informationen beilag, mehr Beispielprogramme hätten in keinem Fall geschadet. Technisch ist der Interpreter gut und bietet sogar Zugriff auf die CLI-Dienstprogramme (dir,type...), die fälschlicherweise als DOS-Aufrufe bezeichnet werden. C-Programme können eingebunden werden, allerdings fehlt hier wie überall sonst auch eine Beschreibung. Mit etwas Aufwand könnte hier eine Sprache für den Amiga entstehen, die eine neue Welt der KI eröffnet.

Forth

Wer gerne in Assembler programmiert, wird Forth sicher lieben, denn es ist der Assembler-Hochsprache am nächsten. Forth ist eine sehr anwenderfreundliche Sprache. Die großen Vorteil von Forth bestehen darin, daß Befehle zu der Sprache addiert werden können. Der Ausdruck:

: hello. .„Hallo Ihr“;

ist der Weg, in Forth einen neuen Befehl „hello“ zu definieren, der, wenn er aufgerufen wird, „Hallo Ihr“ schreibt. Das Programm wird definiert, indem ein Reihe eigener Befehle compiliert werden. So braucht zum Beispiel nur:

hello

eingegeben werden, und der Forth Interpreter führt den neuen Befehl aus. Befehle und Wörter werden in einem „dictionary“, geordnet in verschiedene „vocabularies“ gespeichert. Forth wird manchmal auch „threaded language“ (thread = fädeln, Perlenkettenstruktur) genannt, weil die „vocabularies“ sich durch die „dictionaries“ fädeln und die einzelnen Wörter miteinander verknüpfen. Forth ist nicht die kompakteste Sprache und benutzt „esoterische“ Namen. Erfahrene Forth-Programmierer bestehen auf korrekter Aussprache dieser Namen. Die Fähigkeit, ein Source-Programm in Forth, manchmal sogar sein eigenes, zu lesen beruht auf Klarheit und Stil des Programmierers. Je kürzer und einfacher eine Definition, desto besser — jedes Wort im Programm bedeutet Arbeit. Forth-Befehle leiten Parameter untereinander auf den Stack, doch anders als in anderen Sprachen, sind sie in der Lage, mehrere Ergebnisse an die aufrufende Software zu übergeben. Zwei wichtige Forth-Versionen auf dem Amiga sind CSI Multi-Forth und JForth von Delta Research. Beide sind gut geschriebene, professionelle Softwarepakete mit guter Dokumentation und vielen Beispielen. Beide verfügen über 68000-Assembler-Interfaces und erzeugen selbständige Programme.

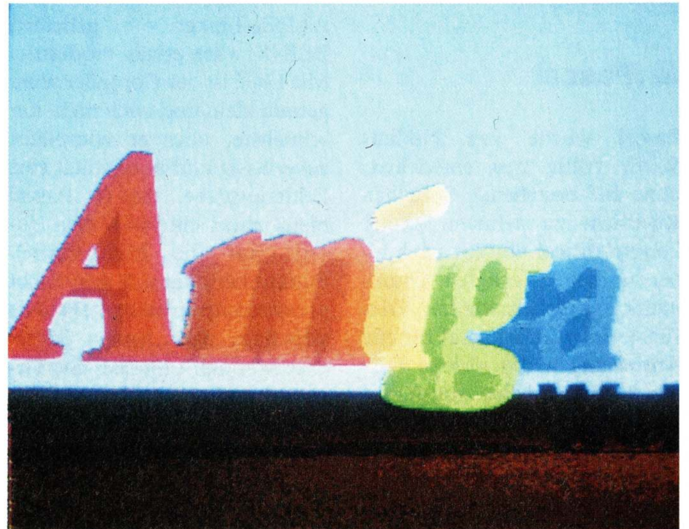
(N. Murray)

Schrift in Licht und Raum

Gelungene 3D-Buchstaben bedurften bislang stundenlanger Bearbeitung. Schneller geht es jetzt mit Fancy-3D-Fonts. Damit erhalten Raytracing-Grafiken eine adäquate Beschriftung.

Wer Beschriftungen für Raytracing-Grafiken im Schnellverfahren selber herstellt, verschandelt seine Bilder mit plumpen, unansehnlichen Lettern. Fancy-3D-Fonts von HS&Y schafft hier Abhilfe. Das Programm enthält drei bekannte Zeichensätze, die der Profi unter den Namen Bookman Italic, Helvetica Bold und Clarendon Medium kennt. Die Buchstaben wurden sorgfältig als dreidimensionale Objekte eingegeben und im Format des Raytracing-Programms Sculpt-Animate abgelegt. Auf den Disketten werden die drei Zeichensätze als „Ital“, „Bold“ und „Fanc“ bezeichnet. Jeder Zeichensatz besteht grundsätzlich aus den gesamten Buchstaben des Alphabets — in Groß- und Kleinschreibweise — und einigen Sonderzeichen wie „Ausrufezeichen“, „Dollarzeichen“, „Eszett“ und „Fragezeichen“. Allerdings enthält nicht jeder Zeichensatz alle Sonderzeichen. Besonders die Umlaute sind vernachlässigt worden. Der Bold-Satz bietet überhaupt keine Umlaute, paradoxerweise jedoch ein „Eszett“. Ital besitzt bis auf das „ö“ alle Umlaute — allerdings, wie bei allen anderen Zeichensätzen dieser Disketten, nur als Kleinbuchstaben. Fanc ist ein leicht verschnörkelter Zeichensatz, der alle kleingeschriebenen deutschen Umlaute kennt, das „Eszett“ und je ein kleines „e“ mit Accent aigu und Accent grave — Frankreich wir kommen. Und benötigt der Anwender gerade die nicht vorhandenen Umlaute eines Fonts, kann der gewünschte Buchstabe zusammengestellt werden.

Die anschließende Verwaltung der Buchstaben ist gut gelöst. In verschiedene Unterverzeichnisse aufgeteilt, findet sich jedes Zeichen als eigene Datei, die als „Scene“ in Sculpt-Animate eingeladen und der restlichen Szene hinzugefügt werden kann. Das Zeichen wird mit Hilfe der Funktion Extrude in allen drei Dimensionen auf die gewünschten Ausmaße gebracht und entsprechend positioniert. Dabei sollte man darauf achten, daß die Buchstaben zumindest in der X/Y-Ebene gleichmäßig verformt werden, damit die Proportionen erhalten bleiben. Perfektionisten müssen dabei beachten, daß ein Zeichensatz ab einer bestimmten Winzigkeit seine Proportionen ändert. Um ihn nicht gänzlich zu entstellen, sollte der Buchstabe jetzt — entgegen den ursprünglichen Proportionen — verbreitert werden.



Den ausgefüllten (gerenderten) Objekten sieht man ihre Vektorstruktur nicht mehr an.

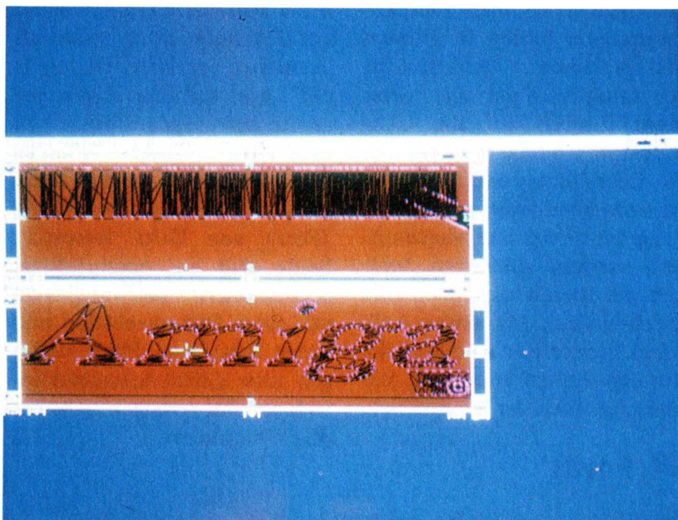
Sculpt-Eindrücke werden unterstützt

Die erstere Version der drei Zeichensätze (auf zwei Disketten) ist ausschließlich für das Sculpt-Animate-Raytracing-Programm gemacht. Möchte man die Zeichen auch in anderen 3D-Grafik-Programmen wie Videoscape 2.0 verwenden, muß man sich entweder nach einem Konvertierungsprogramm umsehen oder auf die Ankündigung des Herstellers vertrauen, daß entsprechende Adaptionen bald erscheinen. Als Konvertierungsprogramm hat sich Interchange von Syndesis bewährt. Damit sind die Zeichen sofort auch für Videoscape, Turbo Silver und Forms in Flight einsetzbar, da für diese Programme zu Interchange Konvertierungsmodule existieren. Erfreulicherweise ist auf der Diskette eine deutsche Übersetzung der englischen Bedienungsanleitung zu finden.

„Fancy-3D-Fonts“ ergänzt Raytracing-Programme sinnvoll. Die Zeichen haben keine Ecken und Kanten. Überflüssige Linien wurden entfernt, so daß die Rechenzeit auf ein Minimum begrenzt ist. Trotzdem darf man die Komplexität der Buchstaben nicht unterschätzen — für die Berechnung des nebenstehenden Bilds in einer Auflösung 640×256, dem Antialiasing-Effekt „Good“ und zwei Lampen benötigt ein Standard-Amiga immerhin etwa sechs Stunden Rechenzeit.

(Ottmar Röhrig)

Info: Heinrichson, Schmeiler & Young, Köln



Im Scriptmodus lassen sich Perspektiven und Größen der Schrift manipulieren.

Drucker mit Plus

Der NEC P6 ist bekannt als leistungsfähiger Drucker. Jetzt kommt der neue NEC-Drukker P6 Plus mit mehr Schriften, mehr Speicher und mehr Möglichkeiten.

Neben einem neuen Design bietet der NEC P6 Plus jetzt einen 80 KB starken Druckerpuffer, was dem Umfang von etwa 50 Din-A4-Seiten entspricht. Der Rechner steht während der Druckzeit anderweitig zur Verfügung. Neu und praktisch ist die Paper-Park-Position, die es erlaubt, auch bei eingespanntem Endlospapier Einzelblätter zu bedrukken. Dazu wird ein kleiner Hebel betätigt, der das Endlospapier so positioniert, daß der halbautomatische Einzelblatteinzug problemlos funktioniert. Soll nach dem Druck auf Einzelblättern wieder auf Endlospapier gedruckt werden, klappt man es mit dem Paper-Park-Hebel in die ursprüngliche Position zurück, ohne sich mit einem langwierigen Einfädeln aufhalten zu müssen. Verwöhnte Anwender können sich sogar einen optionalen automatischen Einzelblatteinzug besorgen.

Sieben Schriften serienmäßig

Am Präzisionsdruckkopf sitzen 24 Nadeln (2 × 12 Nadel versetzt), die ein hervorragendes Schriftbild in allen Druckmodi garantieren. Der NEC P6 plus ist serienmäßig mit sieben einzeln anwählbaren Profischriften ausgestattet (Times, Helvetica, Prestige Elite, ITC Souvenir, Draft Gothik, Bold PS, und Courier), die mit den Sensortasten ausgewählt werden können. Optional einsetzbare Steckkarten ermöglichen weitere Zeichensätze, diverse Spezialschriften (zum Beispiel Super-LQ 360 × 360 Punkte pro Zoll) und Barcodedruck. Die Karten werden in einen Port auf der linken Vorderseite gesteckt. Mit maximal 265 Zeichen pro Sekunde (Draft 12 HS) und 75 Zeichen (LQ 10) erreicht der NEC P 6 plus eine Geschwindigkeit, die bislang nur weit teurere Drucker erreichten.

Es geht auch Bunt!

Grafikausdrucke sind gestochen scharf und werden bis zu einem Grad von 360 × 360 dpi aufgelöst. Da sind selbst anspruchsvolle CAD-Grafiken kein Problem mehr. Wer es bunt mag, kann seinen P6 plus auf Farbdruck nachrüsten. Carbonfarbbänder für ein klareres Schriftbild sind im Fachhandel erhältlich.

Den NEC P6 plus zeichnet seine vollständige Software-Kompatibilität zu allen NEC Pinwritern (24 Nadel-Serie) und anderen Industrie-Standards aus.

Doch NEC produzierte nicht nur technisch Qualität, sondern verfaßte auch ein ausführliches deutsches Bediener-Handbuch, das alle wichtigen Informationen für professionelles Arbeiten enthält. Druckertreiber (inklusive Pinplot) für noch bessere Druckergebnisse und optimalen Druckeinsatz liegen dem Bedienerhandbuch bei.

(Guido Bär)



Seine Qualitäten beweist der NEC P6plus, ohne dabei zu lärmen.

Technische Daten:

Drucktechnologie:
24-Nadel-Präzisionsdruckkopf
2 × 12 Nadeln versetzt
0,2 mm Nadeldurchmesser

Zeilenbreite:
80 Zeichen pro Zeile bei 10 cpi
maximale Einzugsbreite 254 mm

Druckgeschwindigkeit:
Draft 12 HS: 265 Zeichen pro Sekunde
LQ 10: 75 Zeichen pro Sekunde

Auflösung:
360 × 360 dpi

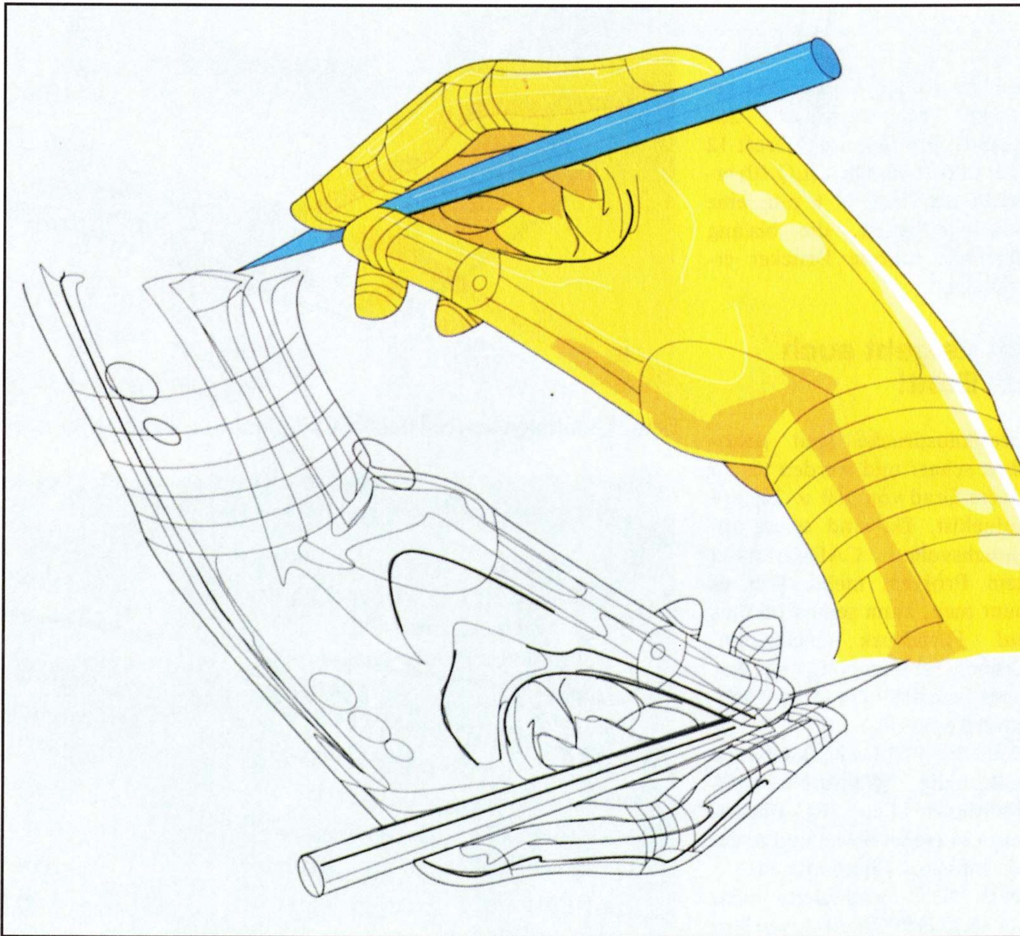
Zeichensätze:
ASCII Standard
IBM Standard/Grafik
Italic
14 internationale Zeichensätze
1 × 256 oder 2 × 128 anwenderdefinierbare Zeichen

Papierzuführung:
Standard: Friktionszuführung
Band-Schubtraktor
Halbautomatischer Einzelblatteinzug
Optinal: Automatischer Einzelblatteinzug

Schnittstellen:
Centronics (Standard)
seriell (optional)

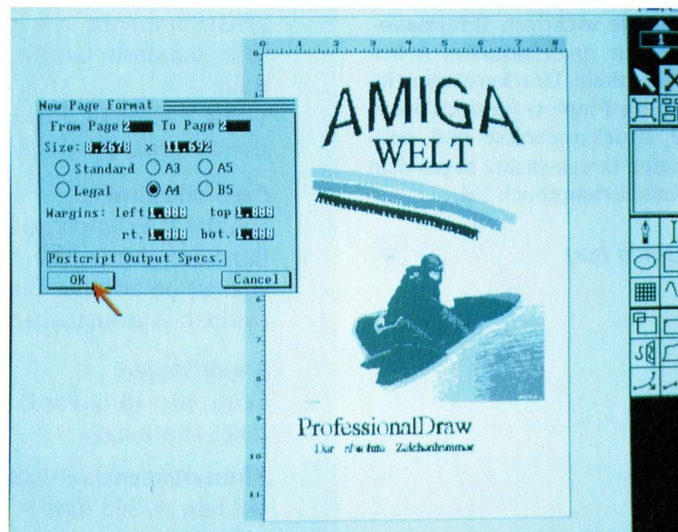
Abmessungen/Gewicht:
440 mm × 360 mm × 145 mm (B × T × H)
9 kg

Mit ProfessionalDraw zum neuen Grafikstandard



Schon das Produktcover signalisiert: Hier geht es nicht um Brushes und Spraydosen, sondern um Linien und Flächen.

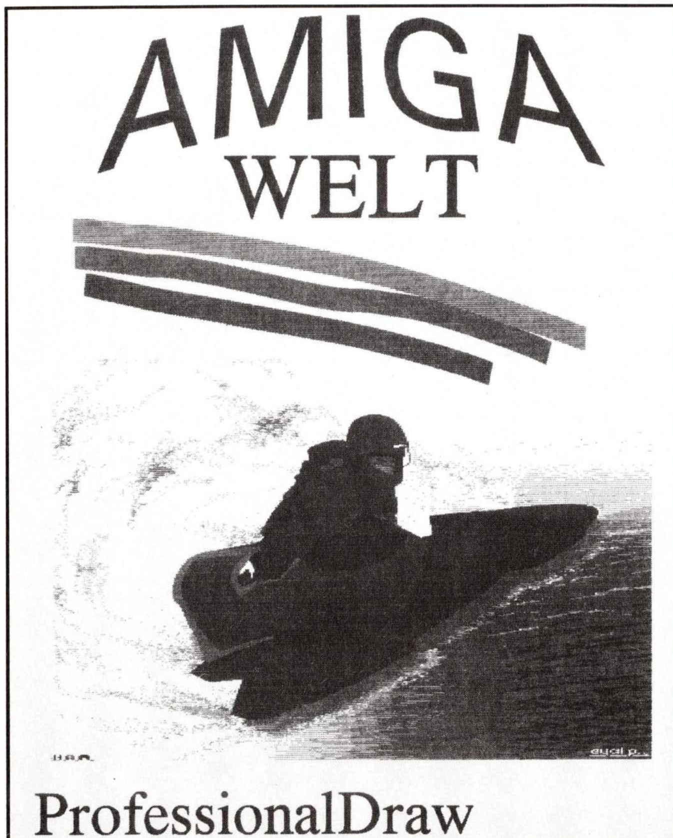
ProfessionalDraw ist ein neues objektorientiertes Zeichenprogramm. Es ist nicht auf technische Zeichnungen beschränkt, sondern dient der Anfertigung von Illustrationen, besonders für Präsentationen und drucktechnische Aufgaben. Mit ProfessionalDraw liegt damit eine Grafiksoftware vor, die dem Amiga Mac-Qualitäten verleiht. Das Rezept dieses hochwertigen Designertools lautet schlicht: Vektorgrafik auf Postscriptbasis.



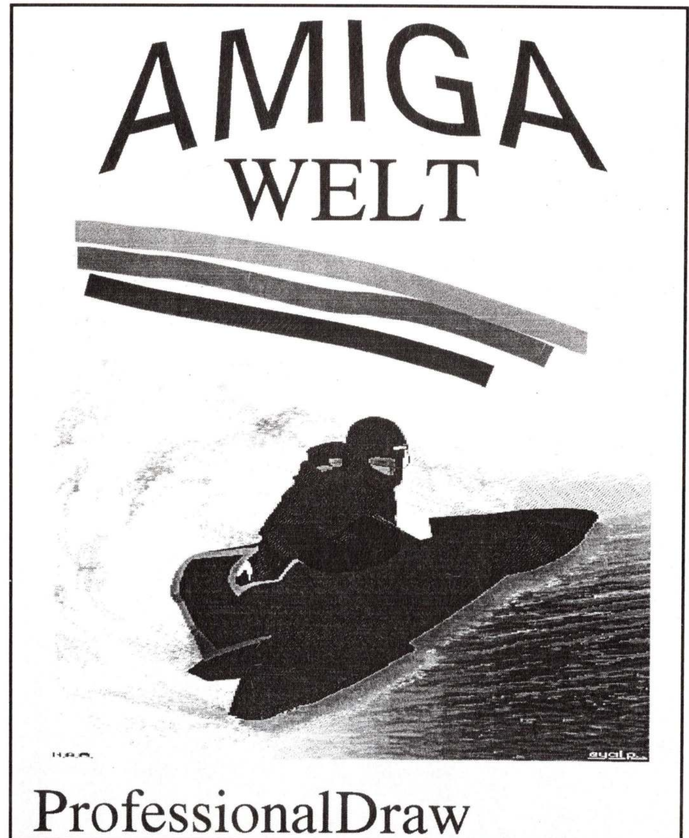
Bedienerfreundlich präsentiert sich der Arbeitstisch.

Gold Disc entwickelte für Amiga das erste Illustrationsprogramm für Postscript-Seiten. Sensationell ist die Tatsache, daß dieses Programm die Postscript-Seiten zeilenweise für Nadeldrucker interpretieren kann. So wird selbst mit einem Low-cost-Equipment professionelles Arbeiten möglich. ProfessionalPage soll mit dem nächsten Update diese Routine implementiert bekommen.

Bei Malprogrammen wie Deluxe PaintII werden die einzelnen Bildpunkte des Bildschirms von Datenregistern repräsentiert. So wird die Grafikfähigkeit des Amiga auf dem Monitor voll ausgeschöpft. Nicht gerecht wird diese Technik den Eigen-



Nur die Farbsättigung des Nadeldrucks . . .



. . . übertrifft der Ausdruck mit dem Laserprinter.

schaften von Druckern, die mit wesentlich höherer Pixelauflösung aufwarten können. Als Resultat erhalten schräge Linien und vor allem Buchstaben Treppchenstrukturen beim Ausdruck. Besonders auffällig ist dieses Manko, wenn in Malprogrammen Zoom-Funktionen verwendet werden. Vergrößerungen verstärken den Treppeneffekt, Verkleinerungen bewirken oftmals ein Aufreißen von geschlossenen Strukturen.

Entwickler aus der PC- und Macintosh-Umgebung haben daher für professionelle Anwendungen im Design- und Druckgewerbe die Seitenbeschreibungssprache Postscript entwickelt. Grafikdaten sind hierbei unabhängig von der Darstellungshardware.

ProfessionalDraw arbeitet mit dieser Standard-Darstellung. Dabei wird alles in Vektordaten dargestellt. Diese legen die Umrisse von Figuren fest. Jedes Objekt besteht aus sogenannten Ankerpunkten, an denen die Verbindungslinien aufgehängt sind. Ein Kreis besitzt seinen Mittelpunkt, an dem man ihn verschieben kann, und vier Ankerpunkte auf der Kreislinie, an denen man ihn vergrößern kann. Ein Recht- oder Vieleck wird mit Hilfe seiner Eckpunkte dargestellt. Kurven und Freihandlinien sind nur allgemeinere Formen von Geraden. Sie besitzen entlang ihrer Bahn mehrere Ankerpunkte, an denen sie nicht nur nachträglich in ihrer Lage und Länge verändert werden können, sondern in jedem Ankerpunkt kann auch mit Hilfe der Tangente in diesem Punkt die Krümmung der Kurve beliebig verändert werden.

Jeder Strich, jede Linie, jedes Vieleck und jede Fläche können nachträglich verändert werden. Es müssen nicht mühsam einzelne Details aus der Grafik herausradiert werden, sondern das betreffende Objekt wird angeklickt und kann dann in den Vordergrund geholt, vergrößert, verkleinert, verzerrt, verschoben oder auch einfach komplett gelöscht werden.

Auch Buchstaben werden als Vektorobjekte dargestellt. Alle Buchstaben sind daher nicht nur beliebig vergrößer- und verzerrbar, wer es darauf anlegt, kann auch den Zeichencharakter an sich verändern. Mehrere Objekte lassen sich zu einer Gruppe zusammenfassen, die gemeinsam weiterverarbeitet werden kann, sich aber auch jederzeit wieder auflösen läßt. Jedem Objekt lassen sich verschiedene Attribute wie Liniendicke, Zeichenfarbe und Füllfarbe zuordnen. ProfessionalDraw bietet alle wesentlichen Funktionen eines Zeichenprogramms, und läßt beim regelmäßigen Gebrauch nur die Füllmuster vermissen (eine typische Eigenschaft von Bitmap-Programmen). Ebenso fehlt die Funktion, um Bitmap-Grafiken an den Umrißkanten automatisch zu scannen, in Vektordaten zu übersetzen, wie es bei vergleichbaren Programmen für den Macintosh üblich ist. ProfessionalDraw bietet nur die Möglichkeit, IFF-Interleaved-Bitmaps einzuladen, um sie entweder als Vorlage zum Nachzeichnen für Grafiker zu verwenden, oder sie als festen Bestandteil in das Layout einzubinden. Bitmapbilder lassen sich in der Größe und im Bildausschnitt nachbearbeiten, alle weiteren Funktionen bleiben den Vektorobjekten vorbehalten.

Für den Refresh des Editorfensters wird eine intelligente Routine verwendet, die nur die veränderten Teile der Grafik löscht und neu aufbaut.

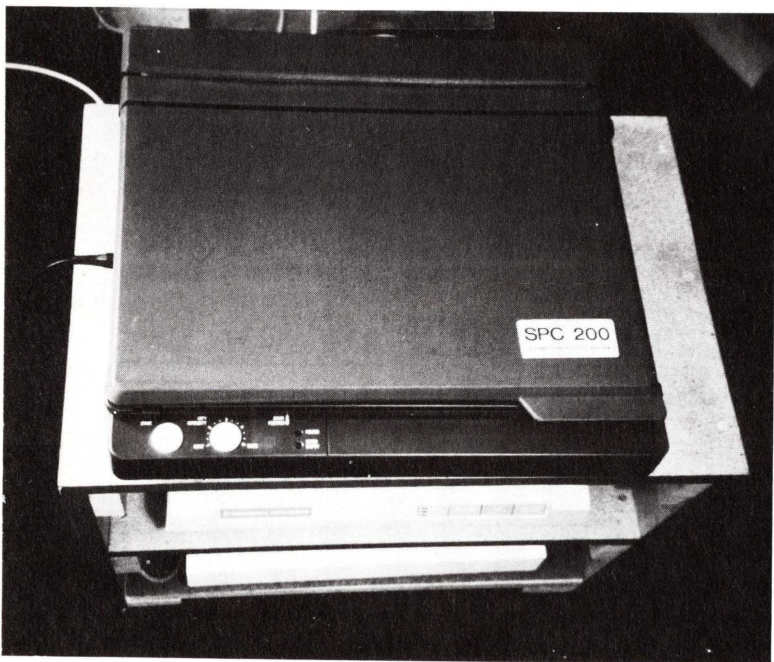
Die Arbeitsgeschwindigkeiten der Routinen sind relativ langsam. Vor allem das Arbeiten mit Text kostet sehr viel Zeit. Auch das Drucken von aufwendig gestalteten Seiten dauert lange (die Demoseite ca. eine halbe Stunde). Dafür hat die Druckausgabe den Vorteil jeden Workbenchdruckertreiber nutzen zu können, und dabei die für den Drucker maximal mögliche Qualität zu erzielen. Die Ergebnisse sind hervorragend.

(Wilfried Häring/uk)

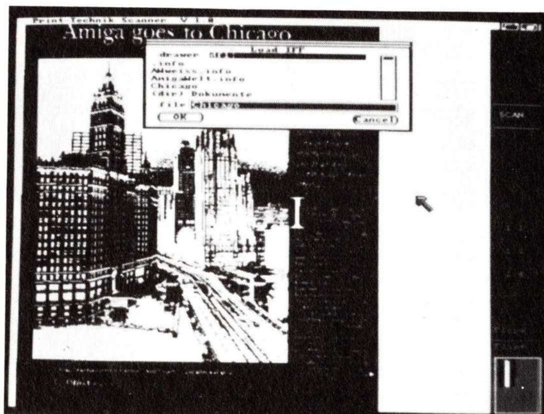
Info: M&T, Haar
Test erfolgte mit einer Pre-Release

Scannen, Drucken und Fotokopieren mit einem Alleskönner

Texte und Grafiken aus dem Computer lassen sich mit einem Drucker zu Papier bringen. Umgekehrt können Scanner den Inhalt eines Blattes in den Speicher einlesen. Mit dem neuen Universalscanner SPC200 läßt sich auf relativ preiswerte Art beides erledigen. Zudem taugt er noch als Fotokopierer.



Drucken, Scannen und Fotokopieren sind die Fähigkeiten des Printtechnik Universalscanner 200



Die gescannte Seite läßt sich mit der mitgelieferten Software editieren

Das haben sich die User schon lange gewünscht: einen Drucker, der auch noch scannen und fotokopieren kann. Der SPC200 von PRINTTECHNIK kann das alles. Das recht preisgünstige Modell gibt es in drei Versionen: für AtariST, Amiga und PC XT/AT. Die Amiga-Version wurde zwei Wochen lang im Alltagsbetrieb getestet.

Bei aller Vielseitigkeit ist der SPC200 auch noch richtig flott. Im Scannbetrieb wird eine DIN-A4-Seite in zehn Sekunden abgetastet. Drucken oder Fotokopieren wird ebenso zügig ausgeführt.

Die Anleitung besteht aus nur sieben Seiten. Dennoch ist alles ausreichend dokumentiert, da die Bedienung wirklich sehr einfach ist.

Der Scanner wird mit dem mitgelieferten Kabel an die parallele Schnittstelle des Amiga angeschlossen. Für den Betrieb sind mindestens 1 MByte RAM erforderlich. Die Software ist autobootend. Soll das Programm von der Workbench aus gestartet werden, so müssen etwa 1,5 MByte zur Verfügung stehen.

Nach dem Start erscheint ein Titelbild mit dem Menü in Form von Gadgets. Zur Auswahl stehen drei Scanner-Versionen:

640×512 Punkte + Gadgets
Wird diese Version geladen, so erscheint ein schwarzer Bildschirm mit einer Menüleiste am oberen Rand und einigen Auswahl-Gadgets auf der rechten Seite. Diese Version läuft im HiRes-Interlace-Modus.

640×512 Punkte
Wie oben, jedoch ohne sichtbare Gadgets. Damit steht der volle Bildschirm zur Verfügung. Bei Bedarf können die Auswahl-Gadgets mit einem Mausklick sichtbar gemacht werden.

640×256 Punkte
Wie die erste Version, jedoch augenschonender, weil „non-interlaced".

Am Beispiel des „Non-Interlaced-Mode" wird die Funktionsweise verdeutlicht. Durch Anklicken des SCAN-Gadgets wird das Scanning gestartet. Allerdings sollte man nicht vergessen, den Schlitten des Scanners ganz nach rechts zu schieben, damit die Software nicht mit einer Fehlermeldung protestiert. Sobald die Vorlage eingescannt wurde, kann in der Darstellung am Bildschirm zwischen drei Arten gewählt werden. Eine A4-Seite wird mit 1760×2048 Punkten gescannt und in einem Puffer gespeichert. Daher der hohe Speicherbedarf der Scan-Software.

Auf dem Monitor ist nur ein Ausschnitt sichtbar

Am Bildschirm kann dann dieses Rohformat dargestellt werden. Im 1:1-Modus besteht die Bitmap aus allen gescannten 1760×2048 Punkten. Auf dem Bildschirm hat allerdings nur ein Sechzehntel davon Platz. Am rechten Bildschirmrand gibt es ein Gadget, mit dem der gezeigte Bereich beliebig verschoben werden kann.

Beim Maßstab 1:2 wird ein Viertel des gescannten Bereiches gezeigt. Das Bild wird auf 880×1024 Punkte komprimiert und in fünf Graustufen dargestellt. Auch hier kann der gezeigte Bereich beliebig verschoben werden. Der 1:4-Maßstab erlaubt, die ganze A4-Seite mit 640×512 Punkten in 16 Graustufen anzuzeigen.

Um bei Halbtonvorlagen in Schwarzweiß- oder mehrfarbi-

AMIGA WELT

Soundfestival

Marktübersicht,
MIDI-Tips,
Public-Domain-Software

Einstieg mit BASIC

Grauwerte werden ebenso hervorragend erfaßt . . .



Amiga goes to Chicago

Nach New York und Los Angeles machte der Amiga-Expreß in Chicago Station. Dabei entlud er hochwertige Güter. Workstations für spezielle Anwendungen machen kostspieligen High-End-Geräten den Markt streitig.

In den Gängen des Nobelhotels „Hyatt Regency" konnte man sich schnell verirren. Letztlich jedoch führten alle Wege in die Ausstellungshallen, in denen sich nach New York und Los Angeles nun zum dritten Mal die Anbieter von Soft- und Hardware den Amiga-Usern stellten. An über 70 Boxen konnten sich die Amiga-Fans mit Informationen und bei genügend Cash auch mit den Produkten selbst eindecken. In einem Land wie den USA sind derartige Veranstaltungen von großer Bedeutung, da kein Händlernetz dicht genug ist, um alle Interessierten ansprechen zu können. Chicago bot daher den Bewohnern des amerikanischen Mittelwestens die einmalige Gelegenheit, sich über die aktuellen Software- und Hardwareapplikationen ▶

Chicago war Gastgeber der dritten „Ami-Expo" — und alles war groß.

6 AMIGA WELT 4/88

. . . wie die kleinen Schriften einer AmigaWelt-Seite

gen Vorlagen optimale Ergebnisse zu erzielen, müssen Bildhelligkeit und Anzahl der Graustufen justiert werden. Die Helligkeit wird mit einem Regler am Scanner selbst eingestellt. Die Auswahl der Graustufen erfolgt mit der Software im SCAN-Menü mit dem Schieberegler „Contrast". Wird der Regler mit der Maus nach rechts geschoben, wird das gescannte Bild kontrastreicher, das heißt, es werden weniger Graustufen verwendet. Stellt man den Regler ganz nach links auf „0", werden 16 Graustufen verwendet. Um optimale Ergebnisse zu erzielen, muß man mit beiden Reglern die beste Einstellung ausprobieren. Nach kurzer Eingewöhnungszeit hat man Erfahrungswerte gesammelt, die Probeläufe überflüssig machen. Wie die Testbilder zeigen, wird eine für diese Preisklasse erstaunlich hohe Qualität erzielt.

Die eingelesenen Vorlagen können sofort ausgedruckt werden. Der Druck erfolgt auf Thermopapier (FAX-Papier). Eine Rolle liefert etwa 100 Blatt A4. Die Anzahl der zu scannenden Zeilen ist auf 2048 voreingestellt. Sie kann fast beliebig verringert werden. Wird zum Beispiel auf 1024 Zeilen umgestellt, so wird nur eine halbe A4-Seite gescannt. In der Option „Drucken" wird dann nur eine halbe Seite (= A5 quer) ausgedruckt. Damit gibt es keinen unnötigen Papierverbrauch.

Daten in Rohformat oder IFF

Die eingelesenen Vorlagen können in zwei Formaten gespeichert werden. Einem IFF-Format, in dem alle gescannten Punkte gespeichert werden, das heißt im Rohformat. Dieses Format kann nur von der Scanner-Software wieder geladen, weiterverarbeitet und ausgedruckt werden. Wichtiger ist jedoch das zweite Format. Hier werden die Bilddaten als IFF-ILBM-Datei mit 640 × 512 Punkten mit vier Bitplanes (= 16 Graustufen) abgespeichert. Dieses Format kann mit allen kommerziellen Grafik-Programmen weiterverarbeitet werden. Die Grafikfiles lassen sich in beiden Formaten auf Datenträgern absichern.

Unabhängig vom Amiga läßt sich der SPC200 als Fotokopierer verwenden. Als sehr praktisch erwies sich, daß die Seitenlänge stufenlos bis zu A4 eingestellt werden kann. Das spart Papier!

Nach zwei Wochen Test- und Alltagsbetrieb erwies sich der SPC200 als völlig störungsfrei. Die Software nahm einige anfängliche Bedienungsfehler manchmal übel.

Mit dem SPC200 erhält der Käufer ein universell verwendbares Gerät zu einem günstigen Preis. Die Software wird ständig weiterentwickelt und für registrierte Käufer gibt es einen Update-Service. Derzeit wird ein Schrifterkennungsprogramm vorbereitet, mit dem Textvorlagen gescannt und als ASCII-File weiterverarbeitet werden können. Zusätzlich ist ein Druckertreiber in Vorbereitung, mit dem der SPC200 (unabhängig von der Scanner-Software) auch als Grafik-Drucker verwendet werden kann.

(rw/iw)

Info:
Der Lieferumfang umfaßt:
- *Universalscanner SPC200*
- *Anschlußkabel für Amiga 500/2000*
- *eine Rolle FAX-Papier (Thermopapier)*
- *Programmdiskette 3,5 Zoll*
- *Bedienungsanleitung*

Preis: ca. 1500 Mark, PRINT-TECHNIK München, 8000 München, Nikolaistraße 2, Tel.: 0 89/36 81 97, PRINT-TECHNIK Wien, A-1060 Wien, Stumpergasse 34—36, Tel.: 02 22/5 97 34 23.

Farbscanner Sharp JX300 für Originalvorlagen

Freihandzeichner müssen auf die Vorteile computerisierter Grafiknachbearbeitung nicht verzichten. Scanner digitalisieren in bester Auflösung papierene Vorlagen. Für Amiga 2000 gibt es jetzt sogar farbtüchtige Hardware.

Findig zeigte sich die amerikanische Firma ASDG. Sie fand einen Weg den Sharp JX300 (DIN A3) oder JH450 (DIN A4), Farbscanner mit einer Auflösung von 300 Punkten pro Zoll und einer Farbauflösung von 16,7 Millionen Farben, über ein Interface mit dem Amiga zu verbinden. Dazu dient eine mit einem speziellen Interfacemodul versehene Multifunktionskarte TwinX für den Amiga 2000. Die Software zum Steuern des Scanners und Einlesen der Grafiken in den Amiga stammt von Perry Kirowitz von ASDG. Der Scanner sieht aus wie ein sehr kleiner Fotokopierer und ist kaum größer als das jeweilige DIN-Standardformat.

Schon der Preis des Gerätes beschränkt die Anwendung auf den absoluten Profibereich. Denn nicht allein der Scanner will bezahlt sein, auch das TwinX-Board mit entsprechendem Modul und der Steuersoftware sind genauso nötig wie mindestens vier MByte RAM im Amiga, besser ist allerdings der Vollausbau auf 8 MByte, eine Festplatte reichlich Kapazität nicht zu vergessen.

Für's erste reicht ein Probelauf

Im ersten Arbeitsgang wird die gesamte Seite für eine Übersicht in wenigen Sekunden schwarzweiß in den Computer eingelesen. Jetzt ist die Auswahl des endgültigen Bildausschnittes möglich; ein beliebig großer (Teil-)Ausschnitt kann für das Scannen in endgültiger Qualität ausgewählt werden.

Als nächstes wählt man die Auflösung beim Scannvorgang zwischen 75 und 300 Punkten pro Zoll und die Farbauflösung des Bildes zwischen den Amiga-üblichen Grafikmodi oder einem Bild mit 24 Bitplanes Tiefe für eine Farbanzahl von 16,7 Millionen Farben. Alle Grafiken können im Interleaved Bitmap Format gespeichert werden. Eine 24-Bit-Grafik mit 300 DPI Auflösung im DIN-A4-Format benötigt allerdings ungefähr 5 MByte Speicherplatz auf der Festplatte. Das Scannen dauert je nach Auflösung und Farbanzahl zwischen einigen Sekunden und zehn Minuten.

Alle Parameter lassen sich bequem editieren

Auf dem Amiga werden die Grafiken auf einem Screen im interlaced RAM-Modus in einer hervorragenden Qualität und höchst möglichen Farbtreue dargestellt. Eine intuitive und attraktive Bedienoberfläche hilft dem Benutzer, die Bilder vielfältig nachzubearbeiten. Die Helligkeit, der Kontrast und die Schwellwerte für den Farbverlauf der drei Fundamentalfarben und Schwarz können nach dem Scannen in der Darstellung verändert werden und das Ergebnis dabei auf dem Bildschirm beobachtet werden.

Außerdem ist es möglich, die Grafik in die vier Farbanteile Schwarz, Blau, Cyan und Magenta aufzusplitten, um sie dann auf einen Postscript-fähigen Druckvorlagenbelichter auszugeben.

Im Moment gibt es nicht allzuviele Malprogramme mit denen sich sehr große IFF-Bilder, die nicht mehr in das Chip-RAM passen, nachzubearbeiten, so wie Deluxe Photo Lab oder Express Paint. Die Grafiken lassen sich dennoch von ProfessionalPage und ProfessionalDraw für den Ausdruck weiterverwenden, solange man in einer niedrigen Auflösung arbeitet. An der vollen Unterstützung von 300-DPI-Grafiken wird laut Gold Disk gearbeitet.

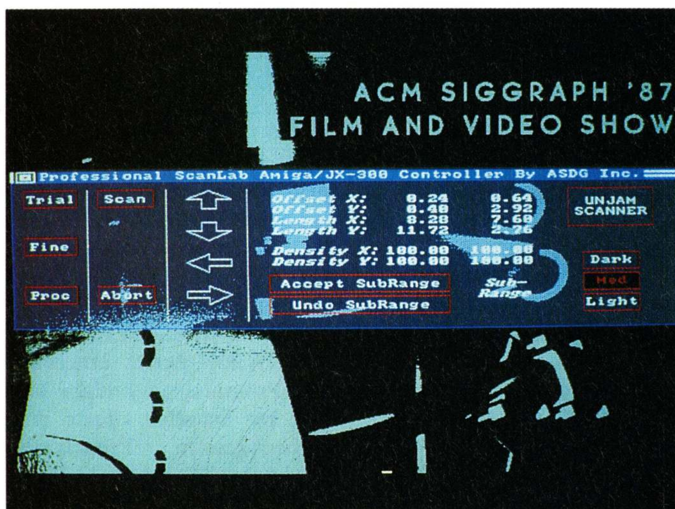
Ein komplettes System für DTP oder Grafikverarbeitung für den Amiga inklusive Scanner, Speicher, Festplatte und der zugehörigen Software stellt fraglos eine hohe Investition dar, und überschreitet die 20 000 Mark. Vergleicht man jedoch die Leistungsfähigkeit mit etablierten Lösungen, zum Beispiel auf Basis des Apple MacIntosh II, sieht man, daß der Amiga ein sehr gutes Preis/Leistungs-Verhältnis bietet.

Bezug: Compustore Fritz-Reuter-Straße 6, 6000 Frankfurt/Main
Scanner inkl. Software und Interface DIN-A4 ca. 15 000 Mark

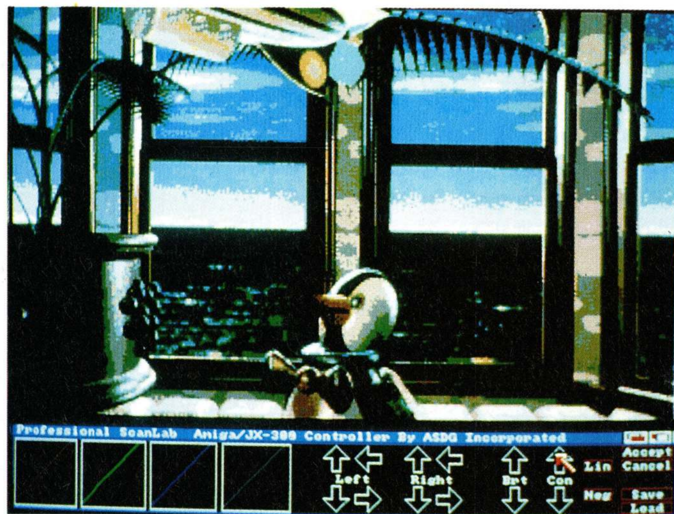
(W. Häring)



Anhand solcher Farbvorlagen kann der Scanner seine Fähigkeiten unter Beweis stellen



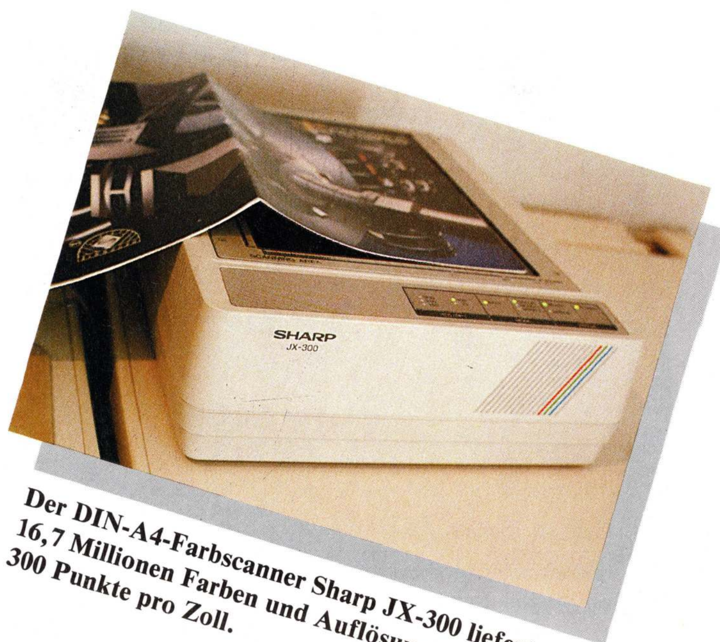
Nach dem ersten Schwarzweiß-Durchgang wird der Bildausschnitt für den Scanvorgang in einem übersichtlichen Requester festgelegt . . .



. . . und ein paar Minuten später ist das Bild für die Weiterverarbeitung in den Amiga eingelesen.



Profi-Arbeitsplatz mit Amiga im Towergehäuse, dem NEC-Multisync und dem brandneuen Sharp-Farbscanner für digitale Bildverarbeitung.



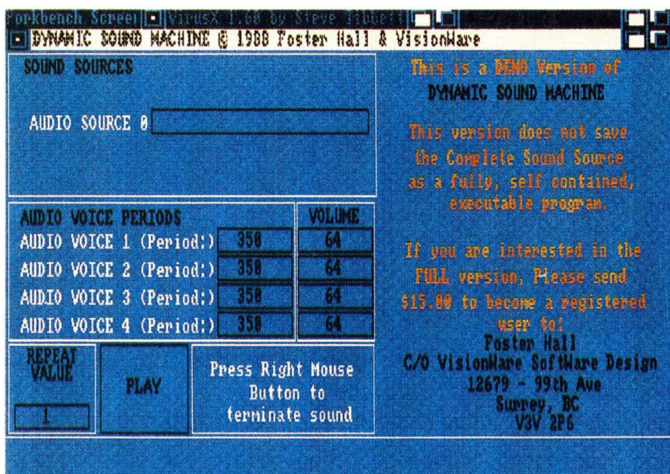
Der DIN-A4-Farbscanner Sharp JX-300 liefert 16,7 Millionen Farben und Auflösungen bis 300 Punkte pro Zoll.

Fish mit Frischesiegel

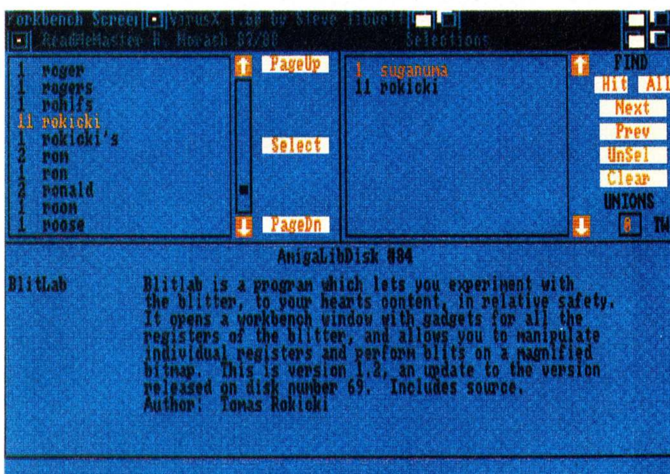
Die Fishdisks 163 – 172



Den neuesten Fishdisks gingen heftige Gerüchte voraus. Von neuen, faszinierenden Versionen der weit verbreiteten Programme Conman und Disksalv war die Rede. Die hochgesteckten Erwartungen wurden nicht enttäuscht.



Von DSM findet sich eine Demoversion auf Fishdisk 167.



ReadmeMaster durchsucht „Contents" nach Stichwörtern.

Fishdisk 163 offeriert mit „Bankn" ein Update zu der Girokonto-Verwaltung von Disk 120, mit „MachII" ein Maus-Utility mit PopCli, CLickToFront und weiteren Features sowie ein Ristinolla-Spiel, dessen Name zugleich die Spielregel charakterisiert: „FiveInLine". „MemTrace" kontrolliert die Belegung und Freigabe des Speichers und ist in erster Linie für Programmierer von Nutzen. Im Verzeichnis „PcPatch" finden sich zwei Programme, die die Befehle PCCopy und PCFormat der Extra-Diskette patchen, so daß sie auf 3¼-Zoll-Formate angewendet werden können. „ReadmeMaster" dient zum Katalogisieren von ReadMe- oder Contents-Dateien und erlaubt ein schnelles Auffinden bestimmter Programme. Mitgeliefert werden die Daten über alle Fishdisks bis Nr. 154. „View" ist ein einfaches Programm zum Betrachten von Texten, das allerdings mit den bisherigen Programmen dieser Kategorie nicht mithalten kann.

Auf **Disk 164** findet man dann auch schon das ersehnte „DiskSalv" in der Version 1.3, die jetzt endlich mit einer vernünftigen Dokumentation versehen ist. Disksalv wird an anderer Stelle in diesem Heft ausführlich besprochen. Auch das bekannte Archivierungsprogramm „Zoo" liegt in einer neuen Version 2.00 vor, die jetzt auch die Chance bietet, Dateien aus einem beschädigten Archiv zu restaurieren. Empfehlenswert ist auch der dateiorientierte Diskettenmonitor „NewZap", mit dem sich zum Beispiel ASCII-Texte innerhalb von Programmen ändern lassen. Im Schatten dieser drei hervorragenden Programme stehen die übrigen: „C-Functions", einige Routinen für Lattice 3.10, „Hed", ein sehr einfacher Texteditor, „PcView" zum Betrachten von IFF-Bildern auf einem EGA-PC sowie zwei CLI-Programme zum Berechnen von Nullstellen in Polynomen: „Newton" und „PolyRoot". In der „PrtDrivers"-Schublade kann man dann noch Druckertreiber für den DEC LN03+ Laserprinter und den Mannesmann TALLY MT420d-Matrixdrucker entdecken.

Neben dem zweiten dicken Knüller, „Conman 1.3", der ebenfalls an anderer Stelle erläutert wird, enthält die AmigaLib**Disk 165** „CPM", die Emulation eines CP/M-Computers mit Z80-Prozessor und ADM3A-Terminal. „Parsnag" und „Separate" unterstützen die Farbtrennung beim Epson JX-80-Drucker, „PlotView" ermöglicht die Darstellung von UNIX-Plot-Formaten auf einem Interlace-Bildschirm. Während „SPUDclock" in wählbaren Zeitabständen die Uhrzeit ansagt, was ziemlich

nervtötend ist, kann „RamCopy" unter gewissen Bedingungen sehr nützlich sein: Dann nämlich, wenn man eine Diskette kopieren möchte und nur ein Laufwerk, dafür aber 1 Megabyte Speicher zur Verfügung hat. Mit „RamCopy" braucht man dabei nur noch einmal die Diskette wechseln. Dazu muß man das Programm in das C-Verzeichnis einer Workbench kopieren, beim Starten des Systems die Startup-Sequence mit Ctrl-D abbrechen und RamCopy aus dem CLI heraus starten. Andere Programme, auch die Workbench, dürfen nicht gleichzeitig laufen, da in diesem Falle auch 1 Megabyte nicht ausreichen würde, um den gesamten Disketteninhalt im Ram unterzubringen.

Auf **Fishdisk 166** befinden sich die Programme „Cref" (zum Erstellen von Cross-Reference-Listen aus C-Sources), „MultiCalc" (Emulation eines Taschenrechners mit umgekehrt polnischer Notation), „Stevie" (Amiga-Version des UNIX-vi-Editors) und „AutoGraf". Letzteres zeigt die Entwicklung von Benzinkosten und -verbrauch über Jahre hinaus und berechnet Durchschnittswerte. Es ist jedoch kein Programm zur Ermittlung privater Aufwendungen für ein Kraftfahrzeug.

Wem es nicht aufwendig genug erscheint, die Syntax der Programmiersprache C zu erlernen, kann sich zusätzlich mit der Syntax von „CDecl" auseinandersetzen, einem Programm, das auf **Fishdisk 167** enthalten ist. Nach erfolgreichem Studium kann er dann Programmzeilen wie
long *(*(*foo) ()) [10]
vergessen. Stattdessen genügt die wesentlich einfachere und verständlichere (?!) Anweisung "declare foo as pointer to function returning pointer to array 10 of pointer to long"
die von „CDecl" in den entsprechenden C-Code übersetzt wird. Um ein Gag-Programm handelt es sich bei „CloseMe". „CLIcon" erlaubt es, CLI-Programme und Batchdateien über Icons aufzurufen, dürfte sich aber nicht gegen das etablierte Xicon durchsetzen können, zumal dieses als IconX auch auf der Workbench 1.3 zu finden ist. „MRPrint" ist ein weiteres Tool zum kontrollierten Ausdrucken von Texten mit Kopfzeilen und Seitennumerierung. Alle übrigen Programme dieser Diskette strapazieren die Audio-Ausgänge des Amiga:

Neben vier hörenswerten „Sounddemos" und einem Demo-Programm für die Erstellung lauffähiger Musikstükke („DSM") erhält man die Version 3.6a von „Smus", die das Abspielen von IFF-Musikstükken ermöglicht. Leider enthält das zugehörige Verzeichnis keine Instrumente.

Die **Disketten 168 und 169**, die eine Einheit bilden, zeigen nicht das gewohnte Bild einer Fishdisk. Sie enthalten die neuesten Versionen und Quellcodes aller Programme, die Matt Dillon bisher der Public Domain zur Verfügung gestellt hat, dazu viele Programmierhilfen und Libraries, die zum Kompilieren der Programme notwendig sind. Aus Platzgründen hat Fred auf eine optische Aufbereitung dieser Disketten verzichtet, deren Inhalt der nebenstehenden Tabelle zu entnehmen ist.

Daß es bisweilen auch recht amüsant sein kann, Programmdokumentationen zu lesen, zeigt **Fishdisk 170.** Hier berichtet Mark Rinfret in der Beschreibung der neuen Version seines Harddisk-Backup-Programms, eine Reihe von Anrufern hätten festgestellt, daß einige Dateien beim Komprimieren länger statt kürzer geworden wären. Mark fand das eigentlich nicht erstaunlich: „Das ist bei mir genauso — immer dann, wenn ich eine Diät in Angriff nehme." Dieser und einige andere kleine Fehler wurden jedoch in der Version 2.4 von „MRBackUp" beseitigt. Der Bug in Marks Diät blieb.

Außerdem bietet die Fishdisk noch die Emulation eines IBM 3278-Terminals („Aftterm"), einen Disassembler („Dis6502) für den 6502-Prozessor (kein Schreibfehler!), eine Assembler-Routine für schnelle Textausgaben („FastText"), ein Grafik-Programm für Bezier-Flächen („Surf"), sowie „Turbo", das umfangreiche Rechnungen beschleunigen kann, indem es den Bitplane-, Sprite, Copper- und Audiozugriff der DMA abschaltet. Am eindrucksvollsten ist jedoch „PtrAnim", mit dem der Mauszeiger animiert werden kann. In jedem Falle lohnt es sich, die mitgelieferten Beispiele einmal auszuprobieren: Einen feuerspeienden Vulkan als Mauszeiger etwa bekommt man nicht alle Tage zu sehen.

Ein wenig spröder zeigt sich dagegen **Disk 171**, die mit „AZ-Comm" ein Terminalprogramm mit ZModem-Protokoll sowie mit „Maze" C-Routinen für zwei- und dreidimensionale Labyrinthe präsentiert. Neben einem System-Monitor („Xoper") findet sich dann noch ein Atari-Freeware-C-Compiler, der allerdings erst noch an Amiga angepaßt werden muß.

Den vorläufigen Abschluß der Reihe bildet die **Fishdisk 172**, die ein für viele C-Programmierer nützliches Tool enthält: „DataToObj" konvertiert Datafiles (Sprites, Bilder, Texte) in Objectfiles, die sich zu einem Programm dazulinken lassen. Die Hiobsbotschaft für Lattice-Anwender: Mit diesem Compiler klappt die Sache leider nicht. Besitzer des Marauder-Kopierprogramms sollten dagegen aufhorchen, denn sie können demnächst um zirka 25 Prozent schneller kopieren. Der wunderschöne Regenbogeneffekt hat nämlich den Nachteil, daß er die übrigen DMA-Zugriffe verlangsamt, was mit dem Programm „MFix" abgestellt werden kann. Weiterhin findet man auf dieser Diskette die Version 1.2 von „ProCalc", einem programmierbaren Rechner, der bereits auf Disk 139 vorgestellt wurde. „Handshake" ist ebenfalls ein Update, und zwar zu dem VT52/100/102/220-Terminalemulator von Fishdisk 60. „Spiff" dient dem Vergleichen zweier Dateien untereinander und lehnt sich stark an das UNIX-Diff-Programm an.
Schließlich gibt es noch „PopInfo" ein kleines Utility für die Workbench, das nach dem Aufruf nur daran erkennbar ist, daß aus dem „W" in der Titelleiste des Workbench-Screens ein „P" geworden ist, die Workbench mithin zu einer „Porkbench". Klickt man das „P" an, so öffnet sich ein Fenster, das Speicher, Errors und Status der im Zugriff befindlichen Disketten, die Speicherbelegung, die Uhrzeit sowie den Charakter der Bootblöcke anzeigt.

(Ralf Leithaus)



Festplatten-Service mal Public Domain

Neuzuweisungen aller Systemverzeichnisse sowie die Datensicherung sind die Hauptprobleme beim Festplattenhandling. Mit Programmen aus der Public Domain lassen sie sich beseitigen.

Mit der Zeit sammeln sich auf der Festplatte immer mehr Dateien an. In manchen Verzeichnissen können es schnell so viele werden, daß der Bildschirm nicht mehr groß genug ist, um sie alle anzuzeigen. List- und Dir-Befehl des AmigaDOS gehen auch recht verschwenderisch mit dem Platz um, indem sie die Dateien ein- beziehungsweise zweispaltig ausgeben. Im Vergleich zu diesen Standard-Befehlen kommt der Eless-Befehl, der sich im c-Verzeichnis der RPD 42 findet, mit weniger Platz aus: Er listet den Verzeichnisinhalt alphabetisch sortiert in vier Spalten. Unterverzeichnisse werden nicht mehr durch die Kennung „(dir)“, sondern durch eine andere Zeichenfarbe kenntlich gemacht.

Zwei weitere Programme sind hilfreich, wenn man einmal den Überblick über seine Verzeichnisstruktur verloren haben sollte. Das Programm Tree aus der CLI-Utilities-Schublade der Fishdisk 162 gibt die komplette Verzeichnisstruktur eines Gerätes aus, wobei sämtliche Unterverzeichnisse berücksichtigt werden. Tree akzeptiert als Argument auch logische Gerätebezeichnungen. Mit dem Befehl

```
tree dh0:
```

läßt sich daher die gesamte Verzeichnisstruktur einer Amiga-Harddisk, mit

```
tree devs:
```

die Struktur des devs.-Verzeichnisses ermitteln. Bisweilen wird es auch vorkommen, daß man eine bestimmte Datei sucht, jedoch nicht mehr weiß, in welchem Verzeichnis man sie abgelegt hat. Die mühsame Suche kann man dem Programm FileFind von Panorama 29c überlassen. Die Syntax dieses Befehls lautet:

```
FileFind Gerät:
[Pfad/]Dateiname
```

FileFind berücksichtigt auch logische Geräte. Als Wildcards zugelassen sind die Zeichenkombinationen „?“, „≠?“ und „*“. Um auf einer jh0:-Festplatte nach der Datei „plain.tex“ zu suchen, eignet sich daher der Aufruf

```
FileFind jh0:plain.fmt
```

Nach kurzer Zeit erhält man das Ergebnis:

```
Found: jh0:tex/macros/
plain.fmt Found 1 match
```

Es wurde also eine solche Datei gefunden, die sich im Unterverzeichnis „tex/macros“ befindet. Mit dem Befehl

```
FileFind jh0:k*
```

wird eine Suche nach allen Dateien auf der Festplatte veranlaßt, deren Name mit dem Buchstaben „k“ beginnt.

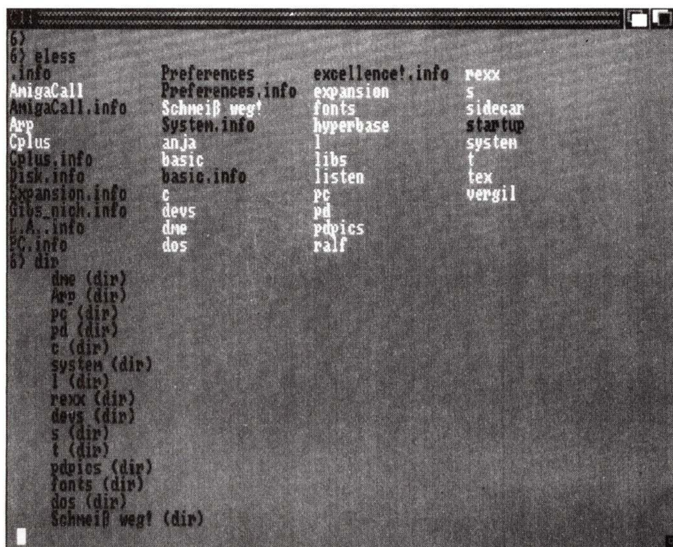
Bei Festplatten, die nicht autobootend sind, muß Amiga nach wie vor mit einer normalen Workbench-Diskette gebootet werden. Um die Festplatte zur System-Diskette zu machen, wird in der Startup-Sequence eine lange Reihe von Assign-Befehlen ausgeführt, um die entsprechenden Verzeichnisse auf der Festplatte als Systemverzeichnisse zu deklarieren. Selbst wenn man die „IF EXISTS . . .ENDIF“-Abfrage entfernt, dauert diese Prozedur sehr lange, da der Assign-Befehl jedesmal neu geladen wird. Dieser Vorgang, dessen Kern aus den folgenden Befehlen besteht,

```
Assign WB: DH0:
Assign c: WB:c
Assign s: WB:s
Assign l: WB:l
Assign libs: WB:libs
Assign devs: WB:devs
Assign fonts: WB:fonts
Assign sys: WB:
```

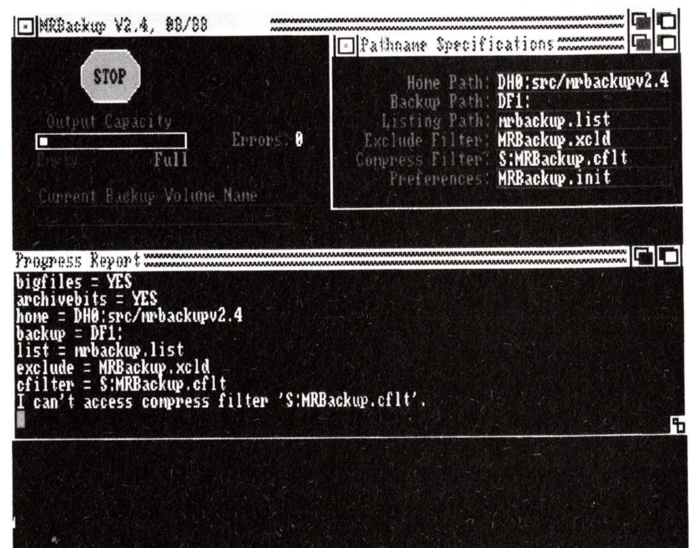
läßt sich in der Startup-Sequence durch eine einzige Zeile

```
dh0:c/defdisk jh0:
```

ersetzen, wenn sich das Programm Defdisk im C-Verzeichnis der Festplatte befindet. Dies ist nicht nur übersichtlicher,



„eless“ bringt Verzeichnisse platzsparend auf den Bildschirm.



Auf Fish 170 findet sich MRBackup zur Datensicherung.

sondern geht auch noch rasend schnell. Defdisk von RPD 118 ersetzt den Assign-Befehl allerdings nur für die Verzeichnisse Sys:, c:, L:, S:, devs:, libs:, und fonts:, von denen man annehmen darf, daß sie auf jeder Festplatte vorhanden sind. Andere Verzeichnisse wie „System" und „Utilities" müssen nach wie vor mit dem Assign-Befehl zugewiesen werden. Auch hier läßt sich allerdings noch Zeit sparen, wenn man auf das Assign-Kommando des ARP (Panorama 27c) zurückgreift, das mehrere Argumente verkraftet und dadurch das Neuladen vermeidet. Die Zuweisung könnte dann lauten:

```
assign System:
dh0:System Utilities:
dh0:Utilities
```

Zu einer weiteren Unannehmlichkeit — bei der Handhabung der Preferences — führt die Neuzuweisung des devs:-Verzeichnisses. Alle Informationen über die in den Preferences getroffenen Einstellungen werden in einer Datei namens „system-configration" gespeichert, die im besagten devs:-Verzeichnis abgelegt wird. In der Startup-Sequence sorgt der Assign- beziehungsweise Defdisk-Befehl dafür, daß AmigaDOS das devs-Verzeichnis der Festplatte als logisches Gerät „devs:" betrachtet. Werden in den Preferences irgendwelche Änderungen vorgenommen, so werden diese also immer auf der Festplatte abgespeichert, was man leicht daran erkennen kann, daß nur das Lämpchen der Festplatte aufleuchtet, wenn man die Preferences über den Save-Button verläßt.

Ganz anders verhält es sich da beim Bootvorgang, wo die Parameter der „system-configuration" ausgewertet werden, bevor die Startup-Sequence abgearbeitet ist. Zu diesem Zeitpunkt liegt für AmigaDOS das devs:-Verzeichnis aber noch auf der Workbench-Diskette, was bedeutet, daß man immer nur die unveränderten Einstellungen erhält, die auf der Workbench-Diskette gespeichert sind. Um die auf der Festplatte gespeicherten Einstellungen zu aktivieren, muß man nach jedem Systemstart erst einmal die Preferences aufrufen. Bequemer ist es da, im hinteren Teil der Startup-Sequence — die Assign-Zuweisungen müssen bereits abgeschlossen sein — die Zeile

```
setprefs devs:
system-configuration
```

einzufügen. Das Programm „Setprefs" (RPD 53) aktiviert auf diese Weise die Einstellungen der Festplatte, ein Effekt der besonders deutlich sichtbar wird, wenn auf Diskette und Festplatte unterschiedliche Farben eingestellt sind. Während der Abarbeitung der Startup-Sequence ist dann ein plötzlicher Farbwechsel wahrzunehmen.

Setprefs kann jedoch auch in anderer Hinsicht für mehr Arbeitskomfort sorgen: Da sich auf einer Festplatte viele verschiedene Programme befinden, ist die Wahrscheinlichkeit hoch, daß diese Programme unterschiedliche Einstellungen benötigen, beispielsweise hinsichtlich der Druckereinstellungen oder der Farbwahl. So wird man etwa für ein Programm, das nur im Interlace-Modus läuft, eine kontrastärmere Farbkombination wählen. Leider können die Preferences aber nur eine einzige Parameterdatei verwalten, so daß man vor dem Aufruf des Interlace-Programms die Farben von Hand ändern muß. Mit Setprefs haben wir dagegen die Möglichkeit, ein ganzes Archiv von Parameterdateien zusammenzustellen. Zunächst legt man sich aber eine Sicherheitskopie der Standard-Einstellung an:

```
cd devs:
copy system-
configuration
config.old
```

Anschließend ruft man mit

```
sys:Preferences
```

die Preferences auf, stellt kontrastärmere Farben ein und verläßt das Programm über den Save-Button. Die neue system-configuration wird nun umbenannt:

```
rename
system-configuration
as Interlace
```

Dieser Vorgang kann für mehrere Einstellungen wiederholt werden. Zum Abschluß sichert man die Standardeinstellung zurück:

```
rename config.old as
system-configuration
```

Die für den Interlace-Modus gewählte Farbkombination kann nun jederzeit mit dem Befehl

```
setprefs devs:
Interlace
```

aktiviert werden. Mit

```
setprefs devs:
system-configuration
```

erhält man wieder die Standard-Einstellung zurück.

Datensicherung mit MRBackup

Obwohl Festplattenlaufwerke eine wesentlich größere Datensicherheit aufweisen als Diskettenlaufwerke, sollte man den Inhalt seiner Festplatte von Zeit zu Zeit sichern. Dazu müssen die Dateien der Festplatte auf mehrere Disketten verteilt werden. Diese Arbeit sollte man einem speziellen Backup-Programm überlassen, von denen es in der Public Domain verschiedene Varianten gibt. Sehr leistungsfähig und von der Workbench aus zu bedienen ist das Programm MRBackup, das in der neuesten Version 2.4 auf der Fishdisk 170 vorliegt. Verglichen mit kommerziellen Programmen wie Quarterback hat MRBackuP jedoch den Nachteil, daß es um ein Vielfaches langsamer ist. Viel Zeit läßt sich aber dadurch ersparen, indem man bereits gesicherte Dateien von MRBackup durch das Setzen eines Archiv-Flags kennzeichnen läßt. Beim nächsten Sichern braucht man das Backup dann nur noch zu aktualisieren, also nur die Dateien zu berücksichtigen, die seit der letzten Datensicherung verändert wurden.

Einen wesentlichen Beitrag zur Datensicherung können im Notfall auch die Programme DiskX und Disksalv leisten, jedoch nur in ihren neuesten Versionen. Disksalv 1.3 von Fishdisk (siehe Besprechung in diesem Heft), ist bis zu einem bestimmten Grade in der Lage, von selbst den Inhalt einer defekten Festplatte, aber auch von jeder normalen Diskette zu restaurieren. Dabei fördert es übrigens so manche längst gelöschten Dateien zutage, wenn auch oft nur bruchstückhaft. Disksalv gehörte auch in seiner früheren Version zur Grundausstattung jeder Public Domain Sammlung.

Im Gegensatz dazu ist der Diskettenmonitor DiskX (Version 2.1 auf Fish 158), der das Editieren jedes einzelnen Bits auf einer Diskette oder Festplatte erlaubt, nur dann verwendbar, wenn man sich genauestens mit der Struktur von AmigaDOS-Disketten auskennt. Ist diese Voraussetzung nicht gegeben, so läßt sich mit diesem Programm nur noch größerer Schaden anrichten. DiskX wurde bereits in Amiga Welt 1/89 besprochen.

Die Installation der hier beschriebenen Utilities dürfte keine großen Schwierigkeiten bereiten, da auf einer Festplatte normalerweise immer noch etwas Platz frei sein sollte. Einige dieser Hilfsprogramme entsprechen AmigaDOS-Befehlen und sind nur vom CLI aus einsetzbar, sie sollten daher in das c-Verzeichnis der Festplatte kopiert werden. Andere Utilities, die auch von der Workbench aus aufzurufen sind, sollten in einem Verzeichnis abgelegt werden, das über ein Schubladen-Icon verfügt. Ein Festplatten-Backup-Programm sollte natürlich auf Disketten aufbewahrt werden, es wird ja gerade dann benötigt, wenn die Festplatte aus irgendeinem Grunde nicht intakt ist.

(Ralf Leithaus)

Backup-Programme: PD gegen Kommerz



Inhalte von Festplatten auf Sicherheitskopien abzusichern, ist ebenso wichtig wie lästig. Daher benötigt der User ein Backup-Programm, das einfach zu bedienen ist und hohe Datensicherheit bietet. Im Vergleich verschiedener Anwenderprogramme können sogar zwei Public-Domain-Kopierer mithalten.

Backups von Festplatten zu ziehen, kostet Zeit und Geld. Ein Blick auf den ergiebigen Public-Domain-Markt offenbart Programme, die nicht nur günstiger in der Anschaffung, sondern teilweise auch einfacher in der Bedienung sind. Im Vergleich müssen sich die kommerziellen Programme „Quarterback" und „Saf-T-Net" gegen die Public-Domain-Varianten „MR-Backup" von der FishDisk 129 sowie „SD-Backup" von Fish 128 behaupten.

„Quarterback", mittlerweile in der vierten Update-Version auf dem Markt, erlaubt nicht nur die Sicherung von Festplattendaten. Jedes Amiga-Device läßt sich abfragen — also auch ser:, par:, ram: oder die neuen Geräte aux: und pipe:.

Ist die Dateiliste eingeladen, kann mittels verschiedener Gadgets ausgewählt werden, welche Dateien gesichert werden sollen. Als besonders nützlich hat sich hier — wie auch bei den anderen Programmen — das sogenannte Archive-Bit der Kickstart 1.2 erwiesen. Normalerweise ist bei jeder AmigaDOS-Datei dieses Bit gelöscht. Es kann jedoch von einem Programm — hier vom entsprechenden Backup-Programm — explizit gesetzt werden. Wird nun die Datei irgendwie verändert oder kommen gar neue Dateien hinzu, werden deren Archive-Bits automatisch durch das DOS gelöscht. Daher müssen beim nächsten Backup-Vorgang nicht wieder alle Dateien, sondern nur die mit gelöschtem Archive-Bit gesichert werden. Quarterback, wie auch die anderen Backup-Programme, bieten diese Option an. Man nimmt allerdings dafür den Nachteil in Kauf, die Dateien in mehreren Portionen vorliegen zu haben. Nur „Saf-T-Net" bietet eine bessere Lösung an.

Sind die zu sichernden Dateien angewählt, hat der Benutzer bei Quarterback die Möglichkeit, einige Parameter einzustellen. So existieren zum Beispiel Schalter für ein Verify der geschriebenen Daten, für einen zusätzlichen Beep beim Diskettenwechsel sowie zum Abschalten der Datenkompression. Als einziger Mitstreiter in diesem Feld bietet Quarterback die Möglichkeit, zwei Diskettenlaufwerke als Ziel anzugeben. Der Backup-Vorgang selbst ist wenig spektakulär. Das Programm teilt dem User ständig mit, wie viele Dateien bereits gesichert wurden und wieviele noch zu sichern sind. Nach vollbrachter Arbeit zeigt Quarterback noch eventuelle aufgetretene Fehler an und kehrt dann zur Backup/Restore-Auswahl zurück.

„Saf-T-Net", in Deutschland noch ohne Vertrieb, kann ebenfalls von jedem AmigaDOS-Device ein Backup anlegen. Im Konfigurationsteil werden alle nötigen Parameter eingestellt. Besonders auffällig sind hier die Gadgets „New Backup Set" und „Append Old Set". Ist Ersteres angewählt, wird — wie bei den anderen Programmen auch — ein komplett neues Backup erstellt. Bei der Anwahl von „Append" — zu Deutsch etwa „Anhängen" — werden dem aktuellen Backup-Block die ausgewählten Dateien angehängt. „Ausgewählt" steht hierbei zum Beispiel für alle Dateien mit gelöschtem Archive-Bit. Durch diese Methode spart man nicht nur Disketten, sondern behält auch einen besseren Überblick über alle Dateien, die sich auf den Backup-Disketten befinden.

Leider bietet „Saf-T-Net" nicht denselben Überblick über den Stand des Backup-Vorgangs wie „Quarterback".

Mit MR-Backup Version 2.1 findet sich auf der Fishdisk 129 ein Programm mit umfangreichen Bedienungsmöglichkeiten. Da gibt es Anzeigen, wie voll die jeweilige Diskette schon ist, Pfade und Verzeichnisse für Backups lassen sich sowohl für das zu sichernde als auch für das Ziel-Device frei wählen. Als besonderen Gag hat MR-Backup eine abschaltbare Funktion, die alle ausgegebenen Texte deutlich ausspricht.

In der Praxis ist jedoch sehr deutlich der Nachteil von MR-Backup zu ersehen. Für die gleiche Datenmenge brauchte Quarterback ca. 3 Minuten, Saf-T-Net das Doppelte und MR-Backup fast das 7fache. 35 Minuten für ein Backup von knapp drei Megabyte ist wohl nur noch für den Sonntag-Nacht-Backup-Job zu vertreten. Das Problem läßt sich hauptsächlich auf die Verwendung des normalen Filesystems der Diskette unter MR-Backup zurückführen. Sowohl Quarterback wie auch „Saf-T-Net" arbeiten nicht, da es nicht dateisondern zylinderorientiert arbeitet. Das hat den Nachteil, daß solche Disketten nur mit diesen Programmen gelesen werden können. Sollte also das Backup-Programm verloren gehen, sind die Daten unbrauchbar. „Quarterback" erlaubt daher auch die langsame Methode über das Standard-File-System.

File für File kopieren bedeutet lange Wartezeiten

„SD-Backup" geht ebenfalls den Weg über das langsame Filesystem. Anders als alle anderen Backup-Programme ist „SD-Backup" ausschließlich CLI-gesteuert. Sämtliche Optionen werden in der Kommandozeile eingegeben. Als einziges Programm arbeitet „SD-Backup" bei den Zielgeräten ausschließlich mit logischen Namen. Absolute Angaben wie „DF0:" sind nicht erlaubt. So können sämtliche Laufwerke des Amiga schon beim Aufruf des Programmes mit den Disketten für das Backup bestückt werden; „SD-Backup" sucht sich dann die entsprechende Diskette selber heraus. Das setzt voraus, das sämtliche Disketten schon vorformatiert sind. Auch SD-Backup unterstützt natürlich wie MR-Backup und die anderen die „Verwertung" des Archive-Bits.

(Ottmar Röhrig)

ConMan von Fish 133 und 165

PD-Power für das CLI

Es gibt noch was Besseres als den Console-Handler NEWCON: ConMan. ist komfortabler und bietet mehr Spezialfunktionen. Geschrieben wurde das Programm von Bill Hawes von dem auch die kommerziellen Programme „WShell" und „ARexx" stammen. ConMan regelt sämtliche Tastatureingaben und ist dafür verantwortlich, daß im CLI das Editieren einer Zeile nun möglich ist.

Auf welcher FishDisk diese Version erscheinen wird, lag zur Drucklegung dieses Artikels noch nicht fest. Die ältere Version 1.1, die der 1.3 aber nur in wenigen Punkten nachsteht, ist auf Fish 133 zu finden. ConMan wird jedoch auch mit einigen kommerziellen Produkten ausgeliefert.

Für die von ConMan benötigten Dateien müssen auf der Workbench Disk ungefähr neun KByte frei sein. Das entspricht 18 Blöcken. Es wird folgendermaßen installiert: Man kopiert die Datei „conhandler-.library" nach „libs:" und „ConHandler" nach „l:". Außerdem sollten die nützlichen Befehle „ConMan", „SetCMan" sowie „History" ins „c."-Verzeichnis kopiert werden. Am leichtesten installiert man ConMan, indem man in der Startup-Sequence an einer beliebigen Stelle „ConMan —c" einfügt. Nun hat sich ConMan als Ersatz für CON: in das System eingehängt, so daß keinerlei weitere Änderungen nötig sind, um bei einem Standard-CLI jetzt die Sonderfunktionen bereit zu haben. Die Option —c stellt den voreingestellten Wert dar und bewirkt, daß sich ConMan als Ersatz für CON: in das System einbindet. Weitere Optionen beim Aufruf von Conman sind:

—bXXXX XXXX gibt die Anzahl der Bytes für den History-Buffer an. Dieser Buffer speichert alle bisherigen Eingaben und hält sie zur erneuten Benutzung bereit. Normalerweise hat der Buffer eine Größe von 488 Bytes. Kommen mehr Befehle hinzu als der Buffer aufnehmen kann, wird die jeweils älteste Eingabe gelöscht.

—t Hiermit wird die „true history" von ConMan eingeschaltet. Die Befehls-History verhält sich hier wie bei den meisten anderen Shells — wie auch bei NEWCON:. Der Zeiger für den letzten Befehl bleibt immer am Ende der Liste stehen. So hat eine Cursor-Fahrt nach unten beim ersten Mal keine Wirkung, weil man sich ja am Ende der History befindet.

—w Mit dieser Option wird die ConMan-eigene „Wrap-Around"-History voreingestellt. Diese History-Behandlung arbeitet nicht mit einem Stack, sondern mit einer sich immer wiederholenden History-Liste.

—lXXXX Hier gibt die Zahl XXXX an, wie lang ein Befehl mindestens sein muß, damit er der History angefügt wird. Die Voreinstellung beträgt 1, ein günstigerer Wert wäre 2.

—o wählt den „Overstrike"-, also Überschreib-Modus an. Hier werden, wenn man mit dem Cursor in die Zeile zurückgegangen ist, neue Zeichen nicht eingefügt, sondern über ursprüngliche Zeichen drübergeschrieben.

—s wählt den „sticky"-Modus an. Normalerweise kann auch während des Betriebs mit CTRL-A zwischen Überschreib- und Einfüge-Modus umgeschaltet werden. Nach einem RETURN ist jedoch immer der voreingestellte Modus aktiv. Soll der Modus auch nach einem RETURN beibehalten werden, ist diese Flag beim Aufruf von ConMan zu setzen.

—r Hiermit wird ConMan anstelle von CON: als RAW: eingebunden; das ist jedoch äußerst selten günstig.

—d Unterdrückt das automatische Senden von CTRL-C an Tasks unter bestimmten Umständen durch ConMan.

Die Optionen —o, —s, —t und —w können auch mit dem Programm SetCMan verändert werden. Der Aufruf mit „setcman ?" gibt eine Liste der möglichen Optionen sowie die gerade angewählten Werte aus. Mit den anderen Optionen können die Werte entsprechend ConMan verändert werden. Es ist zu beachten, daß die mit ConMan und SetCMan eingestellten Optionen global sind, also alle ConMan-(CLI)-Fenster betreffen. Wer nur selten mit dem CLI arbeitet, kann ConMan auch explizit über die Workbench aktivieren. Das geschieht durch je einen Doppelklick auf das „LoadLib"-Icon sowie auf das „ConMan"-Icon. Das CLI ist damit geöffnet und besitzt die ConMan-Funktionen.

Im ConMan-CLI haben einige Funktionstasten und CTRL-Codes neue Bedeutungen bekommen. Zuerst die Funktionstasten:

F1 verkleinert das CLI-Fenster auf minimale Größe. Drückt man F1 erneut, bekommt das Fenster seine vorherige Größe.

F2 vergrößert das CLI-Fenster auf maximale Größe. Ein erneuter Druck stellt den vorherigen Zustand wieder her.

F5 und **F6** durchsuchen die History nach einem bestimmten Befehl. Hat man zum Beispiel weit oben eine sehr lange Link-Anweisung eingegeben, die mit „BLink" beginnt, braucht man nicht mit der Cursor-hoch-Taste alle Zeilen zu durchsuchen. Es wird lediglich „bli" (ohne RETURN!) eingegeben, und der Druck von F6 zeigt die erste Zeile von unten (in Richtung des Anfangs der History) an, die mit dieser Zeichenfolge beginnt. F5 tut dasselbe in Richtung Ende der History.

F7 und **F8** löschen ganze Worte auf Tastendruck. F7 löscht dabei das Wort, auf oder nach dem der Cursor sich gerade befindet in Richtung Zeilenanfang. F8 tut dasselbe in Richtung Zeilenende.

F9 setzt den aktiven Bildschirm in den Vorder- oder Hintergrund, vorausgesetzt, das ConMan-Fenster bleibt währenddessen aktiv.

F10 bewirkt das gleiche am ConMan-(CLI)-Fenster.

Neben den Pfeiltasten, die in Verbindung mit SHIFT das Springen von Wort zu Wort erlauben oder an den Anfang beziehungsweise das Ende der ge- ▶



samten History springen, gibt es weitere Tastenkombinationen. Die CTRL-Codes werden dadurch ausgelöst, daß die CTRL-Taste in Verbindung mit einer anderen Taste gedrückt wird. Hier eine kurze Übersicht:

CTRL-A schaltet zwischen dem Einfüge- und Überschreib-Modus um.

CTRL-^ wählt, unabhängig vom aktiven Modus, immer den Einfüge-Modus an.

CTRL-Y löscht die Zeile von der aktuellen Cursorposition bis zum Ende.

CTRL-U löscht von der Cursorposition bis zum Anfang der Zeile.

CTRL-X löscht die gesamte Zeile.

CTRL-Z löscht die gerade eingegebene Zeile und die Zeilen, die zwar schon eingetippt wurden, aber von ConMan noch nicht gelesen und ausgeführt worden sind.

CTRL- (CTRL-Minus) entspricht dem Undo. Tippt man beispielsweise eine neue Zeile und drückt versehentlich auf eine der Cursor-hoch- oder -runter-Tasten, kann die eigentlich verlorene Zeile mit dieser Tastenkombination wieder aufgerufen werden. Das geht jedoch nur, solange noch kein RETURN gedrückt wurde.

CTRL-S hält — in Anlehnung an die MSDOS-Welt — die Ausgabe auf dem Bildschirm an.

CTRL-Q setzt die Ausgabe auf dem Bildschirm fort.

CTRL-W löscht den Bildschirm und stellt die aktuelle Zeile noch einmal dar.

Es ist sehr übersichtlich, daß alle CTRL- und ESC-Codes, die nicht direkt von ConMan verwendet werden, in der „^ "-Schreibweise ausgegeben werden. Erscheint auf dem Bildschirm also ein „^ K", so muß man zuvor CTRL-K getippt haben. ESC-Sequenzen werden mit einem vorangestellten „^ [" dargestellt. Längere ESC-Sequenzen werden von ConMan solange ohne Ausgabe akzeptiert bis eine unbekannte Sequenz vorliegt. So kann beispielsweise durch Druck auf ESC und einem nachfolgenden [0~ die F1-Taste simuliert werden.

Anhand der Informationen läßt sich mit ConMan arbeiten. Allerdings enthält dieses Programm einige weitere Spezialitäten, die in einer Fortsetzung präsentiert werden könnten.

(Ottmar Röhrig)



TaskControl von Fish 159

Herr über alle Tasks

Alle Freunde des Multitasking werden sich über Fishdisk 159 freuen. Denn sie enthält das Programm „TaskControl", das die Prioritäts- und Statusveränderung laufender Tasks durch die Verwendung einer Intuitionoberfläche sehr vereinfacht.

TaskControl von J. Martin Hippele wird entweder von der Workbench durch einen Doppelklick oder vom CLI aus durch „taskcontrol" oder „runback taskcontrol" gestartet und öffnet sein eigenes Fenster auf dem Workbench-Bildschirm. Gleich nach dem Start wird die aktuelle Taskliste eingelesen und auf dem Bildschirm dargestellt. In der linken Spalte steht für jeden Task eine fortlaufende Nummer, dann dessen Priorität, Name und Status. Am Namen läßt sich gewöhnlich erkennen, um welches Programm es sich handelt. Um nun, während im Hintergrund ein rechenintensiver Bildaufbau läuft, mit einem Texteditor zügig schreiben zu können, setzt man dessen Priorität herauf oder die des Rechenprogrammes herab. Mit TaskControl geht das ganz einfach. Man sucht sich den Task des Editors, beispielsweise „CygnusEd", und bringt ihn mit den Pfeiltasten an der Seite der Liste in das blaue Feld. Nun trägt man in das „Assign Priority"-Gadget die für diesen Task gewünschte Priorität ein, drückt Return und klickt, auf die Frage des Programmes hin, ob dies wirklich auszuführen sei, auf „Do it!". Nun wird die Taskliste erneut eingelesen. Wer sich im letzten Moment anders entschieden hat, kann den Vorgang mit „Cancel" — anstatt „Do it!" — abbrechen.

Vom Benutzer veränderte Prioritäten sollten vom Benutzer nicht unter —10 und nicht über +10 hinaus gesetzt werden. Denn wenn dieser Rahmen verlassen würde, könnte es zu Zusammenstößen mit wichtigen Systemtasks kommen.

Alle folgenden Funktionen müssen zur Aktivierung mit „Do it!" bestätigt oder zur Desaktivierung mit „Cancel" negiert werden. Auch der Status eines Tasks kann verändert werden. Dazu ist der gewünschte Task wieder in das blaue Feld zu bringen. Daraufhin wird der Status zu „Ready", „Wait" oder „Freeze" verändert. Mit „Dead" kann ein Task sogar aus dem System entfernt werden. Darauf sollte man jedoch nur im Notfall zurückgreifen, da eine solche Aktion leicht Systemabstürze verursacht. Ein „gefrorener" Task, der also keinerlei Rechenzeit benötigt und doch im Speicher bleibt, wird im Bedarfsfall mit „Wait" aus seiner Kältestarre geweckt. Beim direkten Anwählen von „Ready" können Probleme auftauchen. Die Taskliste kann mit „List/To Frame" wieder aufgefrischt oder mit „List/Printer" auf dem Drucker ausgegeben werden.

Wird das Fenster durch einen Klick auf das Gadget in der rechten unteren Ecke verkleinert und aus dem Weg geschoben, so steht TaskControl ständig zur Verfügung. Wiederholt man das Ganze, vergrößert sich das Fenster wieder.

(Ottmar Röhrig)

FuncKey von Fish 106

Zehn Tasten — fünfzig Funktionen

Welch eine Verschwendung! Alle zehn Funktionstasten liegen funktionslos brach. Diesem Mißstand hilft das PD-Programm FuncKey ab. Jede F-Taste kann damit maximal fünf Funktionen erfüllen. Die Neubelegung ermöglicht es, das CLI weitgehend über die Funktionstasten zu steuern und simple Texteditoren mit Makro-Fähigkeit auszustatten — und noch mehr. FishDisk 106 enthält das praktische FuncKey 1.1. Diese Version hat zwar kleine Fehler — doch die lassen sich umgehen.

Vor dem Programmstart muß die Datei „FHandler“ in das „c.“-Verzeichnis kopiert werden. Jetzt bieten sich verschiedene Methoden, FuncKey zu starten; jede wirkt sich anders auf das Startverhalten des Programms aus.

Wird FuncKey mit „runback FuncKey“ (runback erschien in der AmigaWelt-Ausgabe 3/88) gestartet, erscheint das Kontrollfeld, in dem die Funktionstasten belegt und andere Einstellungen gemacht werden können. Startet man das Programm mit „runback FuncKey <Dateiname>“, wobei <Dateiname> den Namen einer Datei mit vorher abgespeicherter Funktionstastenbelegung repräsentiert, so erscheint das Kontrollfeld nicht beim Start. Es verbleibt jedoch — unsichtbar — im Speicher und kann mit der Tastenkombination ALT-Help aufgerufen werden. Die Tasten sind jetzt den Dateieinträgen entsprechend belegt. Beim Start von FuncKey über die Workbench verhält es sich ähnlich. Durch Anklicken des FuncKey-Icons wird das Programm gestartet, wobei sofort das Kontrollfeld erscheint. Lädt man FuncKey durch einen Doppelklick auf das Icon einer FuncKey-Datei, werden die Tasten sofort korrekt belegt; das Kontrollfeld erscheint nicht. Im Kontrollfeld selbst gibt es mehrere Gadgets: Eines davon ist „Resident Yes/No“. Damit kann das Kontrollfeld unsichtbar im Speicher gehalten werden. Im Bedarfsfall wird es mit der rechten ALT-Taste und „Help“ aktiviert. Ist das Kontrollfeld nicht resident angelegt, muß es jedesmal durch „Func Key“ oder durch die Workbench eingeladen werden. „FuncKey On/Off“ ermöglicht ein schnelles Abschalten der Funtionstastenbelegung.

In die Zeile „Filename“ wird zuerst der Dateiname eingegeben, unter dem man die Funktionstastenbelegung nach Klick auf „Save“ abspeichern beziehungsweise mit „Load“ laden will. Ist das Gadget „I“ angewählt, wird beim Speichern ein passendes Icon für die Datei angelegt. Das Gadget „Kill“ tut genau das, was es verspricht — es löscht sowohl die Funktionstastenbelegung als auch das eigentliche Programm aus dem Speicher. „Clear“ löscht die Eintragungen für sämtliche Funktionstasten. „Fetch“ entspricht einem „Undo“. Wer aus Versehen „Clear“ gewählt hat, läßt die Belegung mit „Fetch“ wieder aus dem Handler-Prozeß auslesen. „Cancel“ bewirkt, daß sämtliche Änderungen seit dem Aufruf des Kontrollfeldes ignoriert werden. „Use“ sorgt dafür, daß alle Änderungen akzeptiert werden, das Kontrollfeld verschwindet und die Funktionstasten werden belegt. In den oberen zehn Gadgets können jeweils bis zu 79 Zeichen pro Funktionstasten eingegeben werden. Dabei gibt der „Key Qualifier“ an, welche Taste zusätzlich zur Funktionstaste gedrückt werden muß, um das gewünschte Resultat zu erzielen.

Das eigentliche Problem des Programmes liegt nun an dessen Arbeitsweise, die hier kurz angeschnitten werden soll.

Sämtliche Tastatureingaben werden daraufhin untersucht, ob es sich um eine Funktionstaste handelt. Ist dies der Fall, und ist eine Taste belegt, wird der Code für die Funktionstaste verschluckt und die entsprechende Belegung in den Input-Handler Intuitions eingespeist. Dort wird er von dem Filter konvertiert, der durch „Setmap“ gesetzt wurde, weil die Zeichen, die beim Druck auf die Funktionstaste erscheinen sollen, nicht immer identisch mit denen sind, die in die Gadgets einzutragen sind. Diese Regelung betrifft sämtliche Tasten, die durch „Setmap d“ ihre Belegung verändern. Die Tabelle macht das deutlich.

Auf dem Bildschirm soll erscheinen:	In das Gadget muß eingetragen werden:
y	z
:	>
;	<
-	/
/	&
?	_
ü	[
+	]
Ü	{{
*	}}
)	(
=	)
)	*

Diese Belegungen gelten nach einem „Setmap d“. Bei Verwendung anderer Tastaturbelegungen ändert sich die Tabelle. Wie an den doppelten, geschweiften Klammern erkennbar, hält FuncKey einige Sonderfunktionen bereit. Das Zeichen „^“ ersetzt den Druck auf die Return-Taste. „^m“ ist also gleichbedeutend mit „Return“. Um die rechte Amiga-Taste zu simulieren, gibt es die geschweifte, schließende Klammer. Ist die Belegung von F1 mit „}q“ angegeben, bewirkt ein Druck auf F1 genau dasselbe wie ein Druck auf die rechte Amiga-Taste und „q“. Analog dazu verhält es sich mit der geschweiften, öffnenden Klammer und der linken Amiga-Taste. „{n“ würde also linke Amiga-Taste und „n“ simulieren, und damit den Workbench-Bildschirm in den Vordergrund holen. Als letzte Besonderheit erlaubt FuncKey die Simulation bestimmter Intuition-Funktionen. So kann mit der Taste, der die Belegung „\f“ zugewiesen ist, das gerade aktivierte Fenster und mit „\s“ der nächste Bildschirm nach vorn geholt werden. „\w“ aktiviert nacheinander alle Fenster auf dem vorderen Bildschirm — „\W“ holt sie währenddessen nach vorn. Um die Sonderzeichen „\“, „{“, „}“ und „^“ zuzuweisen, müssen sie lediglich zweimal eingegeben werden. „^^a\\“ resultiert aus „^a\“. Bis auf die Konvertierung landestypischer Sonderzeichen bereitet FuncKey keine Probleme. Das Programm ist zuverlässig, absturzfrei und erfordert wenig Speicherplatz.

(Ottmar Röhrig)



SysMon von Fish 58

Gläsernes System

Ergänzend zur resetfesten RAM-Disk von ASDG findet sich im selben Verzeichnis der Fish-Disk 58 das nützliche „SysMon". Es zeigt sowohl den verbleibenden Speicher — getrennt in Chip- und FastMemory — als auch die aktuelle prozentuale Auslastung des Prozessors an.

Das Programm wird entweder mit „runback sysmon" oder von der Workbench aus gestartet und öffnet ein eigenes Fenster. Das Fenster wird so verschoben, daß es nicht bei der Arbeit stört. Für jedes zwei MByte große Fast-RAM erscheint am unteren Rand des Programmes eine Säule. Die Säule für das Chip-RAM repräsentiert allerdings nur 512 KByte Speicher.

Am Balken „CPU Utilization" liest man ab, wie beschäftigt der Rechner gerade ist. Bei komplexen Rechenaufgaben schnellt der Balken bis zum rechten Anschlag; die Auslastung beträgt 100 Prozent. Anhand der „Switch Activity" läßt sich die Rechenintensität aller laufenden Tasks ermitteln. Die angezeigte Zahl gibt die Umschaltungen zwischen verschiedenen Tasks pro Sekunde an. Wieviel Tasks insgesamt im System hängen, geben die Zahlen für „Ready Tasks" und „Waiting Tasks" an. Der Balken „I/O Int/Sec" ist ein Maß für die vom System bewältigten I/O-Vorgänge.

SysMon ist ein sehr nützliches Utility. Lediglich die feststehende Größe des Fensters stört.

DiskSalv von Fish 164

Wenn die Disk irrt

DiskSalv-Autor Dave Haynie hat jetzt die neueste Version 1.3 seines Disketten-Reparaturprogrammes als Public Domain herausgegeben. Nachdem am ursprünglichen DiskSalv einige Verbesserungen vorgenommen wurden, gehört DiskSalv 1.3 nun zu den stärksten Programmen zur Rettung von Dateien auf zerstörten Disketten, RAM-Disk und Festplatten. Sowohl das normale FileSystem des trackdisk.device als auch das neue FastFileSystem der Workbench 1.3 werden ausnahmslos unterstützt. Jetzt können sowohl einzelne Dateien — zum Beispiel nach einem versehentlichen „delete"-Befehl — als auch der Inhalt einer Festplatte auf mehrere Disketten gerettet werden. Die Bedienung ist im Vergleich zur vorhergehenden Version, die keinerlei Optionen bot, ein wenig komplizierter. Die Grundsyntax des DiskSalv-Befehls ist

```
disksalv [from] <Eingabegerät> [to]
<Ausgabegerät/-pfad> [<Opt>]
```

beispielsweise also

```
disksalv df0: to df1:
```

Damit werden alle — auch nur noch teilweise — rekonstruierbaren Dateien der Diskette im Laufwerk df0: auf die Diskette in df1: übertragen. Als Ausgabegerät muß nicht mehr unbedingt ein Gerätename angegeben werden; es sind auch Pfadnamen erlaubt, so daß der Inhalt einer Diskette beispielsweise in ein Unterverzeichnis der Festplatte gerettet werden kann. Als Eingabegerät muß weiterhin ein Gerätename angegeben sein.

DiskSalv bietet jetzt 14 Optionen. Die einfachsten — „disksalv info" und „disksalv help" — geben in dieser Reihenfolge Informationen zum Programm und zum Autor sowie eine Kurzübersicht über die Bedienung des Programms.

Die weiteren Optionen lauten:

`FFS|NOFFS`

Normalerweise muß nicht angegeben werden, um welches File-System es sich handelt. Ist die Diskette jedoch soweit zerstört, daß DiskSalv dies nicht mehr sicher erkennen kann, sollte das Format hier angegeben werden.

`ASK`

Wenn diese Option angegeben ist, wird DiskSalv nach dem Abscannen der Diskette nur die Dateien retten, die wirklich übertragen werden sollen. Das heißt, DiskSalv fragt jeden einzelnen Dateinamen und jedes einzelne Unterverzeichnis ab. Antwortet man auf die Frage mit „Y", wird die Datei gerettet oder — falls es sich um ein Unterverzeichnis handelt — geht DiskSalv in dieses Verzeichnis hinein. Antwortet man mit „N", wird diese Datei oder werden alle Dateien in dem entsprechenden Unterverzeichnis übersprungen.

`NOPRUNE`

Damit wird die sogenannte „Pruning"-Phase übersprungen. Sie stellt sicher, daß auf dem Ausgabegerät normalerweise nur Verzeichnisse eingerichtet werden, die auch Dateien enthalten. Leere Unterverzeichnisse werden automatisch gelöscht. Sollen auch die leeren Unterverzeichnisse generiert werden, ist diese Option anzugeben.

`NOCHECK`

Bestimmte Geräte — wie beispielsweise die RAM:-Disk — sind immer zu 100 Prozent gefüllt, da der Speicher für sie dynamisch alloziert wird. Da DiskSalv den Zustand des Ausgabegerätes stets prüfen muß (eine zu 100 Prozent gefüllte Diskette kann ja nicht weiter beschrieben werden), versagt es bei solchen Geräten und würde ständig eine Fehlermeldung anzeigen. Um DiskSalv einen solchen Spezialfall mitzuteilen, gibt es diese Option. Bei Verwendung des RAM:-Devices wird die Option von DiskSalv automatisch gesetzt.

`NOTD`

Diese Option hat in der vorliegenden Version des Betriebssystems noch keine Bedeutung. Falls es gesetzt wird, unterdrückt es die Spezialbehandlung bei der Rettung von Dateien im trackdisk.device.

`QUICK`

Unterdrückt das Ausgabefenster während der Scan-Phase. Dies erhöht die Arbeitsgeschwindigkeit.

`FORMAT`

Weist DiskSalv an, das Ausgabegerät unbedingt zu formatieren, bevor die erste Datei geschrieben wird. Doch auch wenn diese Option nicht gesetzt ist, kann der Benutzer, falls eine Diskette zu 100 Prozent gefüllt ist, eine neue formatieren. Zu beachten ist allerdings, daß das Formatieren eines Gerätes nur möglich ist, wenn im Verzeichnis „1:" der „Disk-Validator" vorhanden ist.

`START blocknr.|START ROOT`

Mit der Angabe „START" und einer darauf folgenden dezimalen Blocknummer kann DiskSalv dazu angewiesen werden, mit der Untersuchung der Diskette (Scan-Phase) erst beim angegebenen Block zu beginnen. Wird statt einer Blocknummer die Zeichenkette



„ROOT" eingegeben, beginnt der Scan-Vorgang beim Root-Block, der sich gewöhnlich in der logischen Mitte des Eingabegerätes befindet. Das hat bei Geräten mit sehr vielen Blöcken (Festplatten) den Vorteil, daß eine kleine Datei, die kurz nach dem Root-Block abgelegt wurde, sehr viel schneller aufgefunden wird.

```
STOP [+] blocknr.
```

Dies gibt den Endblock der Scan-Phase an. Entweder wird dieser Block als direkte, dezimale Blocknummer oder als Offset vom Startpunkt angegeben. In letzterem Falle muß der Anzahl der zu bearbeitenden Blöcke ein Plus vorausgehen. Die Option „start root stop +100" würde also, angefangen beim Root-Block, die 99 folgenden Blöcke abscannen.

```
MASK
[a|A] [r|R] [w|W] [e|E] [d|D] [p|P] [s|S]
```

Mit dieser Option kann der Benutzer eine Art Filter aufgrund der Schutz-Bits der Datei zwischenschalten. Die Buchstaben entsprechen dabei den bekannten Bedeutungen (Archiv/Read/Write/Execute/Delete/Pure/Script). Wird das Bit in Kleinschreibung angegeben, wird eine Maske aufgelegt, bei der das Bit nicht gesetzt sein darf. Bei Großschrift muß das Bit gesetzt sein. Nach „MASK a" wird DiskSalv also nur die Dateien behandeln, bei denen das Archiv-Bit nicht gesetzt ist. „MASK RW" wird nur die Dateien behandeln, bei denen Lese- und Schreibzugriffe erlaubt sind. Alle nicht angegebenen Bits können in beliebigem Zustand auftreten.
Alle diese Optionen können in einer DiskSalv-Anweisung gemischt auftreten. Wurde DiskSalv gestartet, werden zuerst die Daten des zu bearbeitenden Gerätes angezeigt. Alles o. k.? Dann drückt man RETURN. DiskSalv kann aber auch mit „N" wieder abgebrochen werden. Ein Abbruch mit CTRL-C ist in jeder Phase möglich. Es folgt die Scan-Phase. Dabei wird normalerweise ein kleines Fenster geöffnet, das die Nummer des gerade bearbeiteten Blocks sowie dessen Typ anzeigt. Die Blocktypen File/Root/UDir/Data/Free können mit Dateikopf/Root-Block/Unterverzeichnis/Datenblock/unbelegt übersetzt werden. Zwei weitere Sondertypen ???? sowie ERR! zeigen entweder einen unbekannten Blocktyp oder einen Fehler an. Bei einem Rettungsversuch unter FastFile-System besteht für DiskSalv keine Möglichkeit, zwischen Daten- und unbelegten Blöcken zu unterscheiden. Deshalb wird hier immer ein unbekannter Typ angezeigt! Gleichzeitig zeigt DiskSalv jeden gefundenen Dateieintrag auf dem Bildschirm an und erhöht den „Nodes"-Zähler um den Wert eins.
Bei eingeschalteter ASK-Option werden die Dateien nun einzeln abgefragt. Wird eine Datei gerettet, so wird diese über die normalen Betriebssystemroutinen auf das Ausgabegerät geschrieben. Bei einem Rettungsversuch mit DiskSalv werden die auf dem Eingabegerät befindlichen Daten in keiner Weise verändert. Die defekte Diskette muß nun formatiert werden. Danach können alle Dateien wieder zurückgeschrieben werden. Dabei ist zu bedenken, daß eine Diskette, die einmal ohne Bedienungsfehler einen Error hatte, dies schnell wieder tun kann.
Deshalb ist es besser, solche Disketten gleich auszusortieren und im Falle einer defekten Festplatte die entsprechenden Sektoren zu sperren.

(Ottmar Röhrig)

PS: Autor Dave Haynie bittet darum, Geldspenden für seine Programme nicht an ihn, sondern an Greenpeace zu schicken.

Zoo von Fish 164

Ein Zoo für Amiga

Bei der Datenfernübertragung spielen Programme zur Komprimierung der Daten eine große Rolle. Ebenso ist die Archivierung, also das Zusammenbündeln von Programmen bei der Telekommunikation sehr nützlich. Die einzelnen Dateien werden dabei komprimiert und können mit verschiedenen Kommentaren versehen werden. Die Vorteile bei diesem Verfahren liegen auf der Hand. Eine Übertragung eines Programmes, welches aus mehreren Dateien besteht, geht zügiger voran, da nicht jedesmal wieder ein neuer Dateiname zur Übertragung eingegeben werden muß. Außerdem können sich die Komprimierungsdaten sehen lassen. Kompressionen zwischen 30 Prozent und 50 Prozent sind keine Seltenheit. Rechnet man die verkürzte Übertragungszeit auf die Telefonrechnung um, kann man also mit solchen Programmen bares Geld sparen!
Zoo ist, wie Arc, ein Abkömmling eines Unix-Programmes. Das Datenformat ist auf allen Rechnern und Betriebssystemen gleich. So können geZOOte Dateien auf einer Unix-Maschine gepackt und beispielsweise auf einem MS-DOS-Rechner wieder entpackt werden. Vor allem die gerade erschienene Version 2.0 hat gegenüber der älteren, auf Fishdisk 136 erschienenen Version 1.71 erheblich an Datensicherheit der Archive gewonnen. So können durch Übertragungsfehler zerstörte Archive in den meisten Fällen wieder rekonstruiert werden. Des weiteren sind einige neue Amiga-spezifische Optionen dazugekommen.
Zoo hat gegenüber Arc jedoch noch weitere Vorteile. So können die Namen der Dateien im Archiv beliebig lang sein. Außerdem unterstützt Zoo das Prinzip von Pfaden und Untermenüs des Amiga, was Arc nicht tut. Und als letzter und einer der wichtigsten Punkte ist auch die Kompressionsrate bei Zoo um einige Prozent höher als bei Arc. Zoo arbeitet ausschließlich im CLI. Ruft man Zoo ohne Parameter auf, so erscheint eine Meldung über die möglichen Parameter. Mit "Zoo h" erhält man einen Hilfsbildschirm, der sämtliche Kommandos und Optionen kurz erläutert. „Commandos" sind im Zoo hier mit „Kommandos", die „Modifiers" jeoch mit „Optionen" übersetzt worden. Sowohl Kommandos als auch Optionen sind Parameter des CLI-Befehls.
Eine Zoo-Anweisung kann auf zwei Arten erfolgen. Die einfachste ist die Verwendung der sogenannten „Novice"-Kommandos, den Anfängerbefehlen. Sie erlauben zwar keine derart komplexe Verwaltung des Archivs wie die eigentlichen Kommandos, geben aber eine einfache Einstiegsmöglichkeit in die Verwendung von Zoo. Natürlich sind Archive, die mit den Anfängerbefehlen erstellt wurden, hundertprozentig kompatibel mit den Archiven, die aus den anderen Kommandos entstanden sind. Deshalb möchten wir uns in diesem Teil auch auf die Erläuterung dieser Anfängerbefehle beschränken.
Arbeitet man mit den Anfängerbefehlen, sieht eine Zoo-Anweisung immer so aus:

```
Zoo -<Befehl> <Archivname>
[<Dateiname(n)>]
```

Unter Verwendung der normalen Kommandos und Optionen sieht das schon etwas komplizierter aus:

```
Zoo <Kommando>[<Optionen>]
<Archivname>
[<Dateiname(n)>]"
```

►

Das soll heißen, daß bei der Verwendung der normalen Kommandos auch Optionen folgen können, aber nicht unbedingt müssen. Arbeitet man mit Befehlen, die im Gegensatz zu den einbuchstabigen Kommandos und Optionen immer aus ganzen Wörtern mit einem vorangestellten Minuszeichen bestehen, haben die Optionen keinerlei Wirkung. Wichtig ist, daß zwischen Kommandos und eventuellen Optionen keine Leerzeichen stehen dürfen; ansonsten wird die Option als Archivname interpretiert. Letzteren mußte man übrigens nicht die Endung ".zoo" anhängen — dies erledigt das Programm automatisch. Das Archiv heißt also später auf der Diskette "< Archivname >.zoo". Am einfachsten sind die zehn Anfängerkommandos, die hier in einer kurzen Übersicht dargestellt werden sollen: add, freshen, delete, update, extract, move, print, list, test, comment. Um eine oder mehrere Dateien (hier die Dateien Bild.pic und Text .doc) in ein Archiv aufzunehmen, tippt man:

```
Zoo -add ram:MeinArchiv
Bild.pic Text.doc
```

Dabei bleiben die Einzeldateien jedoch bestehen. Sollen die Einzeldateien gleich nach dem Einfügen in das Archiv gelöscht werden, tippt man statt "add" einfach "move". Besteht das Archiv "ram:MeinArchiv" übrigens noch nicht, wird es neu erstellt — existiert es hingegen schon, werden die Dateien dem Archiv hinzugefügt. Vor allem wenn beispielsweise ein ganzes Unterverzeichnis in ein Archiv aufgenommen werden soll, zeigt sich Zoo von seiner starken Seite. Durch die Verwendung sogenannter „Jokerzeichen" können viele Dateien mit einer einzigen Anweisung übernommen werden. So erzeugt

```
Zoo -add ram:MeinArchiv
dh0:system/*CLI*
devs:*.device
```

ein Archiv mit allen Dateien aus dem Verzeichnis "dh0:system", die die Zeichenfolge CLI aufweisen (NewCLI, CLI, NewCLI.info und so weiter), sowie alle Gerätetreiber aus dem Verzeichnis "devs:". Übrigens wird bei der Verwendung aller Anfängerbefehle der Pfad zu den einzelnen Dateien ignoriert und nicht mit ins Archiv übernommen. Sollte also im folgenden Beispiel eine weitere Datei namens „CLI" in das Archiv übernommen werden, wird die alte Kopie gelöscht. Verwendet man die normalen Kommandos und haben die beiden Dateien „CLI" einen unterschiedlichen Pfad, stören sie sich gegenseitig nicht. Möchte man nun den Inhalt eines Archivs betrachten, hilft der „list"-Befehl. Mittels

```
Zoo -list ram:MeinArchiv
```

erhält man eine Übersicht ähnlich der folgenden Tabelle.

mentar zu einer Datei mittels "/end" oder "/END" in einer neuen Zeile, woraufhin zum nächsten Eintrag gesprungen wird. Soll die Eingabe des Kommentars für alle Dateien abgebrochen werden, drückt man CTRL-N (EOF-Zeichen). Eine schon vorhandene Datei kann mittels „Return" unverändert übernommen werden.

Die Prüfsumme ist nur von Bedeutung, wenn ein defektes Archiv mittels des mitgelieferten Utilities „fiz" rekonstruiert werden soll. Ob die Integrität eines Archivs gewährleistet ist — was zum Beispiel als schnelle Kontrolle nach einer Übertragung interessant scheint — kann mittels des „test"-Befehls überprüft werden. Dazu ist nur ein

```
Zoo -test ram:MeinArchiv
```

angegebenen Pfad liegen und noch nicht im Archiv vertreten sind.

Um eine oder mehrere Dateien wieder aus dem Archiv zu löschen, verwendet man den „delete"-Befehl.

```
Zoo -delete ram:MeinArchiv
CLI* alf.device
```

würde die die Dateien "CLI", "CLI.info" und "alf.device" aus dem obigen Beispielarchiv löschen.

Der letzte und wichtigste Befehl lautet "extract". Er zieht eine Datei wieder aus dem Archiv heraus und speichert sie auf Diskette in ihrer ursprünglichen Form. Werden keine Dateinamen angegeben, werden alle Dateien aus dem Archiv extrahiert. Die Dateien werden nach einem

```
Archive ram:MeinArchiv.zoo:
Length    CF  Size Now  Date      Time
--------  --- --------  --------- --------
    5372  35%     3507  24 Dec 88 15:05:22 4949 C alf.device |Treiber
    6040  28%     4326   1 Jun 88 13:40:22 8a6e   asdg.vdisk.device
    5340  27%     3895   8 Aug 88 15:14:14 ae77   serial.device
    3020  31%     2085   8 Aug 88 15:54:10 ee5b   CLI
     729  60%      292   9 Dec 88 17:13:16 c966 C CLI.info |Icon fuer CLI
    4840  29%     3442  13 Sep 88 10:47:10 131d C PopCLI  |von Fishdisk
--------  --- --------  --------- --------
   25341  31%    17547     6 files
------------
```

Ausführliche Informationen erhält man mit Zoo-list.

In dieser Übersicht werden von links nach rechts die Originallänge der Datei, die Kompressionsrate, die jetzige Länge der Datei, das Datum und die Zeit der Dateibearbeitung sowie deren Checksumme (CRC) angezeigt. Dann folgt ein C, falls der Datei ein Kommentar zugeteilt ist (der bei den Anfängerbefehlen auch gleich angezeigt wird) und der eigentliche Dateiname eventuell mit dazugehörigem Pfad (siehe oben). Ein Kommentar kann mittels des „comment"-Befehls an eine Datei angehängt werden. Nach

```
Zoo -comment ram:MeinArchiv
```

werden nacheinander alle Dateinamen aufgelistet. Hier kann dann zu jedem einzelnen Eintrag ein Kommentar von bis zu 64 KByte Länge eingegeben werden. Beendet wird der Kom-

notwendig. Eventuelle Fehler werden sofort ausgegeben.

Für die ständige „Pflege" des Archivs gibt es zwei weitere Befehle, die viel Arbeit sparen. So wird nach

```
Zoo -freshen ram:MeinArchiv
dh0:system/*CLI*
devs:*.device
```

jede Datei im Archiv durch eine eventuell existierende neuere Version ersetzt. Diese zeichnet sich dadurch aus, das deren Dateidatum neuer als das der Datei im Archiv ist. Die alte Datei-Version im Archiv wird dabei durch die neue überschrieben. Ist auf der Diskette keine neuere Version der Datei zu finden, geschieht überhaupt nichts. Anders beim „update"-Befehl. Hier werden, im Gegensatz zu „update", auch die Dateien ins Archiv übernommen, die im

```
Zoo -extract ram:
MeinArchiv
```

im aktuellen, mit dem CLI-Befehl "cd" angewählten, Verzeichnis abgelegt. Ist also nicht viel Speicher vorhanden, kann das Archiv hilfsweise auf die Disketten kopiert werden. Nach

```
copy ram:MeinArchiv
df0:MeinArchiv
cd ram:
Zoo -extract df0:MeinArchiv
```

stehen alle Dateien in der RAM-Disk. Die komprimierten Kopien der Dateien im Archiv werden durch diesen Schritt in keiner Weise verändert.

(Ottmar Röhrig)



DMouse von Fish 160

Adrenalin für die Maus

DMouse1.09 von Fish 160 macht die Maus perfekt. Das 11 KByte kleine Programm bietet Hervorholen und Zurückdrängen von Fenstern und Bildschirmen, beschleunigte Mausbewegung, automatisches Schwarzschalten des Bildschirms, Löschen des Mauszeigers, Aufrufen eines CLI über eine Tastenkombination und vieles mehr.

Zuerst wird die Datei „DMouse-Handler" in das „1:"-Verzeichnis kopiert. Nachdem man „DMouse" (ohne „run" oder „runback") aufgerufen hat — meist aus der Startup-Sequence heraus — wird dieser Handler in das System gehängt. Beim Aufruf des Programms können viele Optionen übergeben werden. Alle Besonderheiten von DMouse können auf Wunsch ein- und ausgeschaltet werden. Das Löschen des DMouse-Handlers mittels „dmouse quit" oder das nachträgliche Modifizieren bestimmter Werte kann aus dem CLI heraus geschehen. Die Optionen für „dmouse" lauten im CLI wie folgt (# steht hier als Platzhalter für eine dezimale, $ als Platzhalter für eine vierstellige hexadezimale Zahl):

-quit löscht DMouse aus dem Speicher.

-h zeigt einen Hilfsbildschirm mit kurzer Erläuterung aller Optionen.

-a# verändert die Beschleunigung der Maus. Der voreingestellte Wert beträgt 3, wird -a1 angegeben, ist die Option abgeschaltet.

-t# bietet eine Möglichkeit der Mausbeschleunigung. Hier wird angegeben, um wieviel Pixel sich der Mauszeiger pro Signal von der Maus bewegen soll. Der voreingestellte Wert beträgt hier 0, was das Abschalten der Option bewirkt.

-s# bestimmt die Anzahl der Sekunden, in denen keine Tastatur- oder Mausaktion erfolgen darf, damit der Bildschirm auf schwarz geschaltet wird. Der voreingestellte Wert beträgt 300 Sekunden.

-m# bewirkt das gleiche wie -s, nur daß jetzt der Mauszeiger betroffen ist. Der voreingestellte Wert 5 besagt, daß der Mauszeiger nach 5 Sekunden Bewegungslosigkeit verschwindet und bei Berührung der Maus sofort wieder erscheint. So umgeht man das mitunter recht lästige Verdecken von Texten in einem Editor.

-c# bestimmt die Anzahl der Klicks mit der linken Maustaste, die benötigt werden, um das Fenster nach vorne zu holen. Der voreingestellte Wert ist hierbei 1.

-p# setzt die Priorität des DMouse-Handlers. Der eingesetzte Wert sollte immer größer als 50 sein. Voreingestellter Wert ist 51. Diese Option läßt sich nicht mehr verändern, wenn DMouse erst einmal läuft. Eine Angabe ist nur beim ersten Programmstart möglich.

-w# 0 = verwendet die Amiga-Funktion UpFrontLayer(), um Fenster nach vorne zu bringen. 1 = verwendet WindowToFront(), was sogenannte Simple-Refresh-Fenster korrekt behandelt. Das kann jedoch zu einem Absturz führen, wenn gleichzeitig ein Disk-Icon auf der Workbench angewählt ist. Dies ist ein Fehler im Betriebssystem, der auch in Version 1.3 noch auftritt.

-L# 1 = ermöglicht das Hervorholen eines Fensters durch Klicks mit der linken Maustaste. 0 = schaltet diese Möglichkeit ab.

-A# 0 = Fenster wird niemals automatisch aktiviert. 1 = Fenster wird aktiviert, wenn sich der Mauszeiger darüber befindet. 2 = Fenster kann durch Tastatureingabe aktiviert werden. 3 = vereinigt 1 und 2.

-1$ gibt diejenige Taste an („Qualifier"), die zusätzlich zur linken Maustaste gedrückt werden muß, wenn ein Fenster in den Vordergrund geholt wird. Eine Liste über mögliche Einträge ist weiter unten zu finden. Der voreingestellte Wert 0 bewirkt, daß keine zusätzliche Taste gedrückt werden muß.

-R$ gibt den Qualifier (siehe -1) für die Funktion an, die das Fenster in den Hintergrund bringt. Der voreingestellte Wert 4000 gibt an, daß zusätzlich zur rechten Maustaste, die diese Funktion eigentlich auslöst, die linke Maustaste gedrückt werden muß. Ein -R0003 würde die Funktion nur aufrufen, wenn die rechte Maustaste zusammen mit einer der Shift-Tasten gedrückt werden würde.

-K$ setzt den Tastaturcode der Taste, die zusammen mit dem Qualifier aus -Q$ gedrückt werden muß, um das später einzugebende Kommando auszuführen (normalerweise wird damit das CLI oder die Shell aufgerufen). Der voreingestellte Wert beträgt 0045 und repräsentiert die ESC-Taste.

-Q$ gibt den Qualifier zu -K$ an. Voreingestellt ist mit 0040 die linke Amiga-Taste.

-C setzt das nach -K$ und -Q$ auszuführende Kommando. Voreingestellt ist der Befehl „NewCLI". Diese Option muß stets als letzte Option in der Zeile angegeben werden, da alle verbleibenden Zeichen als Kommandozeile aufgefaßt werden.

Die erwähnten Qualifier stellen die Modifikationstasten des Amiga dar. In der folgenden Liste werden alle möglichen, sinnvollen Einträge angegeben.

0001 Linke Shift-Taste
0002 Rechte Shift-Taste
0004 Caps-Lock-Taste
0008 CTRL-Taste
0010 Linke Alt-Taste
0020 Rechte Alt-Taste
0040 Linke Amiga-Taste
0080 Rechte Amiga-Taste
2000 Rechte Maus-Taste
4000 Linke Maus-Taste

Alle Qualifier können auch kombiniert werden. Addiert man die entsprechenden Hex-Werte, kann die gewünschte Funktion durch jeden der Qualifier in Zusammenhang mit der entsprechenden Maustaste oder der Kommandotaste ausgelöst werden.

(Ottmar Röhrig)

MC1K von Fish 36

Sekunde um Sekunde

Mit dem Programm „MC1k", einer Abwandlung des Programmes „clock" von FishDisk 36, steuert der deutsche Programmierer Christian Balzer ein äußerst nützliches Utility für die Workbench der PD-Sammlung bei. Wenn es aus der Startup-Sequence heraus gestartet wird, sollte der Befehl „runback" aus AmigaWelt 3/88 vorhanden sein. Auf den Befehl „runback mclk" erscheint in der Kopfzeile des Workbench-Bildschirms eine sekündlich weiterschreitende Uhr und die Anzeige über noch verbleibenden Speicher — getrennt nach Chip- und Fastmemory. Die Uhr überlagert die eigentliche Speicheranzeige der Workbench und kann mit den Close-Gadgets wieder verlassen werden. Insgesamt benötigt das Programm rund 20 KByte Speicher und eignet sich daher eher für Amigas mit Speichererweiterung. Die Uhr läuft mit einer Priorität von 20. Das macht sie so genau und raubt doch kaum CPU-Zeit.

ASDG-RAM-Disk
von FishDisk 58

Resetfeste RAM-Disk einfach installiert

Frischgebackene Amigabesitzer haben vielfach Probleme, komplexe Programme ordentlich zu installieren. Dieser Artikel soll Anfängern eine Anleitung der ASDG-RAM-Disk von Fish-Disk 58 in die Hand geben, damit sie auf solch ein nützliches Utility nicht verzichten müssen. Führen Sie die folgenden Schritte ganz genauso aus, wie sie hier aufgelistet sind. Die Bedienung der RAM-Disk und der Utility-Programme wird dann später erläutert.

Die Installation ist wesentlich einfacher, wenn dem System mindestens ein externes Laufwerk bekannt ist. Die Anleitung geht davon aus, daß die Fish-Disk Nr. 58 in df1:, also dem externen Laufwerk liegt, und das System mit einer Standard-Workbench (Sicherheitskopie! Kein Original) der Version 1.2 oder 1.3 gebootet wurde. Ist kein zweites Laufwerk vorhanden, muß überall in den folgenden Befehlen „df1:“ durch „AmigaLibDisk58:“ ersetzt werden. Dann sind jedoch wesentlich mehr Diskettenwechsel erforderlich.

Zuerst muß auf der Workbench etwas Platz geschaffen werden, um die zusätzlichen Dateien für die ASDG-RAM-Disk aufzunehmen. Dazu sind etwa 20 KByte nötig. Das Beste ist, wenn zuerst die „Demos“-Schublade gelöscht wird. Dazu ist diese einmal anzuklicken und dann im Menü mittels „Discard“ zu löschen. Wird mit einer Workbench-Version 1.3 gearbeitet, existiert diese Schublade nicht. Hier ist es vorteilhaft das Notepad im Utilities-Verzeichnis auf die gleiche Art und Weise zu löschen. Nun muß ein CLI- (v1.2) beziehungsweise Shell-(v1.3) Window geöffnet werden. Das dazugehörige Icon befindet sich entweder im System-Verzeichnis (v.1.2) oder im Hauptverzeichnis (v1.3). Bei v1.2 kann es vorkommen, daß dieses Icon nicht sichtbar ist. Dann sollte man zuerst das Preferences-Programm einladen, das Gadget für „CLI On“ anklicken und Preferences mit „save“ wieder verlassen. Öffnet man das System-Verzeichnis erneut, müßte das Icon erscheinen. Im CLI tippt man dann

```
copy df1:asdg-rrd/asdg.vdisk.device devs:
copy df1:asdg-rrd/#?RamDisk#? sys:system
join devs:mountlist df1:asdg-rrd/mountlist as
ram:komplett
delete devs:mountlist
copy ram:komplett devs:mountlist
```

Nun muß die Startup-Sequence noch soweit verändert werden, daß die RAM-Disk auch angemeldet wird. Dies geschieht wie folgt:

```
ed s:startup-sequence
```

Nun tippt man „mount vd0:“ und — falls v1.2 Verwendung findet — auch noch „dir >nil: vd0:“. Beide Zeilen werden jeweils durch Return getrennt, so daß sich die Startup-Sequence um zwei Zeilen (v1.2) oder eine Zeile (v1.3) verlängert.

Jetzt sollte die maximale Größe der RAM-Disk noch an die spezielle Systemkonfiguration angepaßt werden. Dies bezieht sich hauptsächlich auf den Speicherausbau. Diese Änderungen werden in der mountlist vorgenommen, die wiederum mit

```
ed devs:mountlist
```

editiert werden kann. Hier sollte der Eintrag „vd0:“ mittels der Cursor-Tasten gesucht werden. Bei Version 1.3 ist hier nun zusätzlich — am Besten als letzte Zeile vor dem Nummernzeichen („#“) — die Zeile „Mount = 1“ einzufügen.

Ohne Speichererweiterung (512 KByte) sollte der Eintrag „BuffMemType = 3“ auf keinen Fall geändert werden. Der Wert hinter „HiCyl“ darf in einer solchen Konfiguration den Wert 15 nicht überschreiten. Das gibt eine maximale RAM-Disk von 128 KByte, was nicht heißt, daß dieser Speicher dem System gleich abgezogen wird. Vielmehr ist das die absolut größte Ausdehnung, die die ASDG-RAM-Disk erreichen kann — dann ist sie zu „100 % full“. Anders sieht das alles bei vorhandenen Speichererweiterungen aus. In folgender Tabelle können sie jeweils die Maximalwerte für den „HiCyl“-Eintrag finden. Diese sollten nicht überschritten werden, damit die Sicherheit bei einem Reset gewährleistet ist.

Gesamtspeicher	maximaler Wert für HiCyl	maximale Größe der ASDG-RAM-Disk
1 MByte	31	256 KByte
1,5 MByte	95	768 KByte
2,5 MByte	191	1,5 MByte
4 MByte	255	2 MByte

Ist eine Speichererweiterung vorhanden, muß der Eintrag des „BufMemType“ auf 5 geändert werden! Theoretisch wären natürlich auch RAM-Disks mit mehr als 2 MByte denkbar. Nur unterstützt die Version des RAM-Disk-Treibers von Fish 58 solch große RAM-Disks nicht. Die ASDG-RAM-Disk wird wie ein normales Diskettenlaufwerk mit dem Gerätenamen vd0: angesprochen. Lesen und Schreiben von Dateien erfolgt über dieselben Routinen wie für alle anderen Laufwerke, so daß in kommerziellen Programmen im FileRequester als Pfad einfach vd0: eingegeben werden kann.

Stürzt der Rechner einmal ab, sind die Daten in der RAM-Disk nicht verloren. Bootet man erneut mit derselben Diskette, wird die RAM-Disk vd0: mit all ihren Daten wiederhergestellt. Nur eine Stromsperre überlebt diese RAM-Disk nicht. Die beiden Utility-Programme „DeleteRamDisk“ und „CleanRamDisk“, die sich jetzt im System-Verzeichnis der Workbench befinden, haben besondere Aufgaben. Nach einem DeleteRamDisk (man kann es der Einfachheit halber in „drd“ umbenennen) wird die RAM-Disk nicht mehr resetfest und wird beim nächsten Reset komplett gelöscht. CleanRamDisk („crd“) gibt nach einem Aufruf alle nicht benötigten Spuren der RAM-Disk wieder frei, woraufhin dem System wieder mehr Speicher zur Verfügung steht. Normalerweise wird dies vom vd0:-Treiber in bestimmten Zeitabständen automatisch erledigt, möchte man alle „Pflege“-Routinen des Treibers jedoch explizit aufrufen, ist CleanRamDisk vom CLI oder der Workbench aus das Richtige. Die resetfeste RAM-Disk vd0: von ASDG benötigt, ohne das ein Bit Daten in ihr gespeichert sind, zirka 25 KByte für den Treiber, die erste Spur und so weiter. Kopiert man nun Daten in die RAM-Disk hinein, werden immer ganze Spuren der RAM-Disk vom Speicher abgezogen — dieser verringert sich also in Schritten von 8 KByte.

Nun hat jeder Benutzer endlich ein „Gerät“, mit dem er nach Herzenslust experimentieren kann, weil nichts kaputt geht, und die Daten nach einem Reset trotzdem nicht verloren gehen — ein wahrhaftes Prunkstück der Public-Domain-Szene, das wir dem Autor Perry S. Kivolowitz verdanken.

(Ottmar Röhrig)



PtrAnim von Fish 170

Und die Maus bewegt sich doch!

Mauszeiger gibt es mittlerweile in allen denkbaren Formen und Farben. Seltener findet man animierte Mousepointer, die ihr Erscheinungsbild fortlaufend ändern. Mit PointerAnimator von Fishdisk 170 soll die Workbench belebt werden.

Neben dem unbestrittenen Unterhaltungswert, den man schon beim Anklicken der Beispiele in der PtrAnim-Schublade registriert, kann ein geschickt animierter Mauszeiger durchaus auch sinnvolle Aufgaben erfüllen: Er kann zum Beispiel die Aufmerksamkeit auf bestimmte Ereignisse lenken, hebt sich aber auch deutlicher vom Bildschirmhintergrund ab. Der letzte Aspekt wird um so bedeutsamer, je weniger Farben ein Screen besitzt. Der Pointer-Animator besteht aus drei Einzelprogrammen: „merge“ zum Erstellen neuer Pointer, „pa“ zum Aktivieren der Pointer und „KillPointer“ zum Wiederherstellen des gewohnten Mauszeigers.

„PA“ ruft fertige Pointer auf

Bereits erstellte Mauszeiger lassen sich mit Hilfe des Programmes „pa“ aufrufen. Dieses Programm ist auf feste Werte für die Ablaufgeschwindigkeit der Animation, die Höhe des Mauszeigers in Bildpunkten sowie die Lage des sogenannten „hot spot“ voreingestellt. Der „heiße Punkt“ ist derjenige Bildpunkt (Pixel), der beim Anklicken von Objekten die tatsächliche Lage des Mauszeigers (beim Standard-Pointer die Spitze des Pfeils) bestimmt. Da diese Werte nicht für alle Animationen geeignet sind, können sie überschrieben werden. Aus dem CLI heraus kann man sich dazu folgender Optionen bedienen.

—s<n> Diese Option bestimmt die Geschwindigkeit, die auf 50/n Bilder pro Sekunde eingestellt wird. Je größer n, desto langsamer wird die Animation. Ein Wert von 1 ergibt die Maximalgeschwindigkeit von 50 Bildern/s. Voreingestellt ist ein Wert von 12, der also 10 Bildern pro Sekunde entspricht.

—h<n> Die Höhe des Mauszeigers wird in Pixeln festgesetzt. Erlaubt sind Werte zwischen 1 und 255, die Voreinstellung beträgt 16 Pixel. Falsche Einstellungen führen zu einem laufenden Bild, wie man es von falsch justierten Fernsehgeräten her kennt.

—x<n> Der waagerechte Abstand des hot spots zur linken oberen Ecke des Bildes wird in Pixeln festgelegt. Voreingestellt: 0

—y<n> Der senkrechte Abstand des hot spots zum linken oberen Bildpunkt wird festgelegt. Dieser Wert ist ebenfalls auf 0 voreingestellt.

Um diese Optionen an dem „Shock“ genannten Mauszeiger einmal auszuprobieren, begibt man sich in das CLI und wechselt mit dem cd-Befehl in das Verzeichnis „PtrAnim“. Jetzt kann man diesen Mauszeiger durch Eingabe von

pa shock

mit den Voreinstellungen des Programms aktivieren und feststellen, daß diese Werte für „Shock“ recht gut geeignet sind: Am Mauszeiger züngeln kleine Flämmchen empor. Die Wirkung der verschiedenen Optionen läßt sich am besten durch Experimentieren erlernen: „pa shock —s1“ läßt die Flammen wesentlich schneller emporzüngeln, „pa shock —h32“ zeigt den Mauszeiger in doppelter Ausfertigung, „pa shock —h37“ versetzt ihn in eine vertikale Bewegung. Mit dem Befehl „pa shock —y100“ wird der hot spot um 100 Pixel nach unten versetzt, ein Wert der zwar nicht praktikabel ist, die Wirkung dieser Option jedoch besonders deutlich zeigt.

Die Kommandozeilen-Optionen sind natürlich nutzlos, wenn man einen Mauszeiger von der Workbench aus aufrufen möchte. Hier muß man die jeweilige Datei mit einem PROJECT-Icon versehen, indem man zum Beispiel die Datei „shock.info“ kopiert, und ihr den Namen der gewünschten Mauszeiger-Datei gibt:

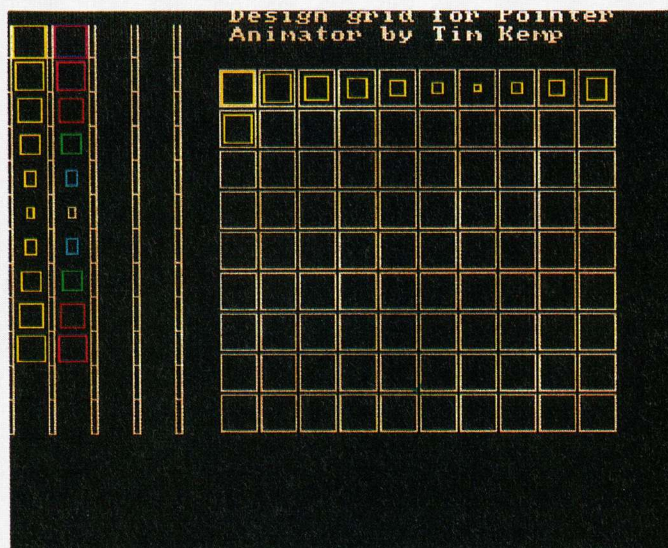
```
copy shock.info
<MeinZeiger>.info
```

Nach einmaligem Anklicken dieses neu erzeugten Icons muß nun der Info-Befehl aus dem „Workbench“-Menü aufgerufen werden. Hier gibt man im DEFAULT TOOL-Eingabefeld den Programmnamen „pa“ einschließlich seines Pfades an. Unter TOOL TYPES können jetzt die neuen Werte für die Animation spezifiziert werden, wozu die folgenden Schlüsselworte zu benutzen sind: SPEED, HEIGHT, X OFFSET, Y OFFSET. Ein Eintrag kann also folgendermaßen aussehen:

```
SPEED=1
HEIGHT=37
Y OFFSET=100
```

Die Handhabung des Info-Befehls ist übrigens am Ende der Notepad-Beschreibung im Amiga-Benutzerhandbuch nachzulesen.

Neben guten Ideen, einem Konzept sowie einem gewissen zeichnerischen Talent, benötigt man für die Erzeugung von animierten Mauszeigern ein Malprogramm, das in der Lage ist, ▶



Mit einem DPaint-Designfeld werden eigene Maus-Kreaturen designed

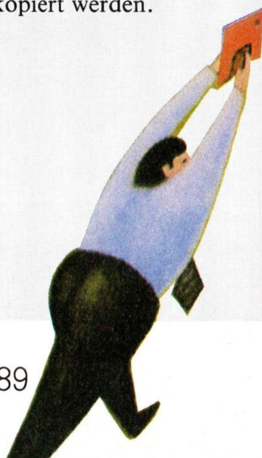
Bildausschnitte (Brushes) in vier bis 16 Farben abzuspeichern. Das verbreitetste dieser Programme dürfte DeluxePaint sein. Von Vorteil ist es auch, wenn das Programm eine Lupenfunktion sowie eine Koordinatenanzeige besitzt, da es auf pixelgenaues Arbeiten ankommt. Als Koordinatenanzeige für den Mauszeiger kann notfalls auch das Programm „Surveyor" von Fishdisk 151 herangezogen werden.

Die Animationen mit „Ptr Anim" arbeiten nach dem Prinzip des sogenannten Daumenkinos. Jeder dürfte die kleinen Heftchen kennen, die bei schnellem Durchblättern, den Eindruck bewegter Figuren hinterlassen. So tut auch „Pa" nichts anderes, als in schneller Abfolge verschiedene einheitlich große Bilder zu zeigen. Wenn sich jedes Bild geringfügig von seinem Vorgänger unterscheidet, entsteht der Eindruck einer flüssigen Bewegung. Bei der Erstellung dieser Einzelbilder sollte man sich Funktionen des Malprogramms wie das Kopieren oder Rotieren von Brushes zu Nutzen machen.

Als Beispiel wollen wir eine einfache Animation mit Hilfe des Programmes DeluxePaint erstellen. Bevor wir das Malprogramm starten, legen wir auf einer beliebigen Diskette ein Verzeichnis an, in dem wir unsere Arbeitsergebnisse für die weitere Bearbeitung ablegen. Ich nenne es „Pointers". Am besten kopiert man in dieses Verzeichnis gleich das „DesignGrid", das sich ebenfalls in der „PtrAnim"-Schublade befindet. Nun kann man DPaint starten, LoRes mit 16 Farben anwählen und das „DesignGrid" laden. Wie man sieht, handelt es sich um ein vorgefertigtes Raster für Bilder in der Standard-Mauszeigergröße von 16×16 Pixeln. In der rechten Hälfte des Rasters kann man nun seine Einzelbilder erstellen. Unser Beispiel ist sehr schnell zu bewältigen, es handelt sich um ein gelbes Quadrat, das kleiner und größer wird. Die Hintergrundfarbe von DPaint, das ist die erste Farbe in der Palette, sollte nicht geändert werden, da nur sie auf der Workbench durchsichtig bleibt.

Da „Pa" die Einzelbilder zyklisch abarbeitet, sollte man dafür sorgen, daß die Einzelbilder am Ende wieder in den Ausgangszustand übergehen. Unserer Abbildung kann man mehr entnehmen, als sich in Worten ausdrücken läßt. Die Einzelbilder müssen nun als Brushes ausgeschnitten werden und in der linken Hälfte des Rasters, zwischen die langen, vertikalen Markierungen abgelegt werden, wobei wieder auf sorgfältiges Positionieren zu achten ist. Auf diese Weise erhält man eine oder mehrere „Bildersäulen", die in dieser Form von dem Programm „merge" verarbeitet werden können. Auch diese Säulen werden als Brushes herausgeschnitten und dann in das „Pointer"-Verzeichnis abgespeichert, beispielsweise unter dem Namen „Column1". Beim Ausschneiden ist unbedingt darauf zu achten, daß die Höhe des Brushes ein Vielfaches der gewünschten Pixelhöhe ergibt. In unserem Fall sind es genau 160 Pixel, die 10 Einzelbildern entsprechen. Für einen weiteren Versuch läßt sich der Vorgang in ähnlicher Weise wiederholen, nur daß die Quadrate vor dem Abspeichern unterschiedliche Farben erhalten. Als Dateiname bietet sich jetzt zum Beispiel „Column2" an.

Das Programm „merge", das unsere Bildsäulen in fertige Mauszeiger verwandelt, sollte nun in das c-Verzeichnis der Arbeitsworkbench oder direkt in das Pointers-Verzeichnis kopiert werden.



Nachdem man mit dem cd-Befehl in das Pointers-Verzeichnis gewechselt ist, genügt der Befehl

```
merge Zeiger1 column1
```

und der fertige Mauszeiger wird unter dem Namen „Zeiger1" im „Pointers-Verzeichnis" abgelegt. Ausprobieren kann man ihn mit dem Kommando

```
AmigaLibDisk170:PtrAnim/pa
Zeiger1
```

In analoger Weise verfährt man mit der Datei „Column2". Man kann aber auch beide Bildsäulen hintereinander verarbeiten lassen:

```
merge Zeiger3 column1
column2
```

Das Ergebnis ist ein Quadrat, das abwechselnd einen gelben und einen bunten Zyklus durchläuft. Man kann beim Aufruf eines Mauszeigers auch eine Information ausgeben lassen.

Diese Information legt man in Form einer Textdatei im „Pointers"-Verzeichnis ab, etwa unter dem Namen „Infortext", und teilt „merge" mittels der „-t"-Option den Namen dieser Datei mit:

```
merge Zeiger1 -tInfotext
column1 column2
```

Beim Aufruf des Mauszeigers mit „pa" erscheint dann zunächst ein Requester, der zusammen mit dem Text eine Copyright-Meldung ausgibt. Erst nach Anklicken des „OK"-Feldes wird der neue Mauszeiger aktiviert. Möchte man einen Mauszeiger automatisch in der Startup-Sequence aufrufen, sollte man auf diese Option verzichten.

Eine Breite von mehr als 32 Pixeln ist jedoch nicht erlaubt. Die Höhe darf maximal 255 Pixel betragen. Das gesamte Programmpaket ist Shareware. Die Adresse des Autors lautet:

Tim Keep
P.O. Box 23101
Columbus, OH 43223

(Ralf Leithaus)

noKlickStart von RPD 66

Schallschutzmaßnahme für Laufwerke

„noKlickStart" von der RPD 66 schont die Nerven all jener, die mehrere Laufwerke besitzen. Jetzt ist mit dem ständigen Rattern leerer Laufwerke Schluß. Die auf der Diskette 66 vorhandene Version modifiziert die Kickstart 1.2 so, das ein Klikken nicht mehr auftritt. Trotzdem werden Diskwechsel noch richtig erkannt und dem System gemeldet. Leider funktioniert das jedoch nicht mit allen Laufwerken. Vorbedingung ist ein Laufwerk, mit einem eingebauten Track-00-Sensor. Fremdlaufwerke wie das NEC-1036 oder dessen Varianten sowie die TEAC-Laufwerke besitzen diese Lichtschranke. Nur ist bei den meisten Amigas ein Laufwerk eingebaut, welches diese Besonderheit nicht besitzt.

Der Trick des Programms besteht darin, den ständigen Kopfwechsel von Track 00 auf Track 01 und zurück durch ein Rückwärtssteppen zu ersetzen. Hat das Laufwerk den erwähnten Track-00-Sensor, steppt es nicht weiter nach hinten, verursacht somit kein Geräusch. Nach dem Start von noKlickStart muß die Kickstartdiskette eines Amiga1000 in das interne Laufwerk geschoben werden. Daraufhin wird die Korrektheit der Version überprüft und die Kickstart entsprechend verändert. Das sollte man nur mit einer Sicherheitskopie machen, da eine Rückwandlung in eine normale Kickstartversion nicht mehr möglich ist.

Besitzer eines A500 oder A2000 müssen sich die so modifizierte Kickstart noch auf Eproms brennen.

Ganz frisch auf dem Public-Domain-Markt ist ein noKlickStart-Version zur Modifizierung der Kickstart 1.3. Die Funktionsweise ist jedoch vollkommen identisch.

(Ottmar Röhring)

Wenn Sie Ihren PC stets gezielt nutzen wollen, liefert Ihnen die PC WELT jeden Monat die nützlichsten Informationen:

Jetzt eine Ausgabe kostenlos zum Probelesen!

9/SEPTEMBER 88 7,50 DM

Das Computer-Magazin für den professionellen Anwender

PC WELT

Kaufberatung:

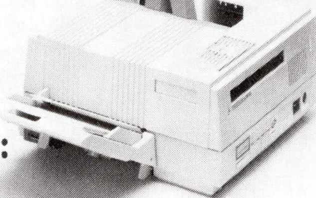
Laserdrucker unter 10 000 Mark

Tests: Grafiksoftware für Profis ● Alle aktuellen Spiele ● Neue Tools für Jeden ● Compaq-386/25 ● Wechsel-Festplatte ● Ergonomie: Besser arbeiten am Computer ●

Neu: DOS 4.0 Mit Maus und mehr Speicher

20 Laptops im Vergleich

ISSN 0175-0496



Preisrätsel: Tulip PC compact 2 zu gewinnen

PC WELT ist die monatliche Fachzeitschrift für den professionellen PC-Anwender.
Redaktionsschwerpunkt sind alle Anwendungen, die auf Personal Computern unter den Betriebssystemen PC-DOS, MS-DOS, BS-2, OS-2, Xenix oder Unix laufen. Ausführliche Testberichte der Hard- und Software liefern wichtige Hintergrundinformationen und vermitteln die notwendigen Kenntnisse über das Angebot im PC-Markt.

Für gezieltes PC-Know-how: PC WELT!

Unser Angebot:
Eine aktuelle Ausgabe PC WELT kostenlos zum Kennenlernen. Nach Erhalt haben Sie 10 Tage Zeit, PC WELT zu prüfen. Überzeugt Sie die PC WELT, brauchen Sie nichts weiter zu tun. Sie erhalten das Magazin dann künftig regelmäßig. Wenn Sie an einem Abonnement nicht interessiert sind, genügt eine kurze Mitteilung an den Verlag, und es erfolgt keine weitere Zusendung.

CW PUBLIKATIONEN

Bitte heute noch absenden an: CW-Publikationen GmbH, Vertrieb, Postfach 40 04 29, D-8000 München 40

Ihr Informationsvorsprung:
☐ JA, senden Sie mir die PC WELT im Rahmen Ihres Angebots. Nach Erhalt meiner Gratisausgabe habe ich 10 Tage Zeit mich zu entscheiden. Hören Sie nichts weiter von mir, schicken Sie mir das Magazin bitte für mindestens ein Jahr im Abonnement. Inlandspreis: DM 80,— (12 Ausgaben) Auslandspreis DM 90,—/sfr. 88,—. Das Abonnement verlängert sich um ein weiteres Jahr, wenn es nicht acht Wochen vor Ablauf des Bezugsjahres schriftlich gekündigt wird.

Garantie: Ich weiß, daß ich diese Vereinbarung innerhalb von 10 Tagen bei der Bestelladresse schriftlich widerrufen kann, wobei bereits die rechtzeitige Absendung meines Widerrufschreibens zur Fristwahrung ausreicht. Ich bestätige das durch meine zweite Unterschrift.

Datum/Unterschrift

Meine Anschrift: ☐ Geschäftsadresse ☐ Privatadresse

Name/Vorname

Firma (falls Lieferanschrift)

Straße, Hausnr./Postfach

PLZ Ort

N

Beruf, Funktion

Gewünschte Zahlungsweise — bitte ankreuzen

☐ ***Bequem und bargeldlos*** durch Bankabbuchung.

☐ ***Gegen Rechnung,*** zahlbar sofort nach Erhalt (Bitte Rechnung abwarten, keine Vorauszahlung leisten).

Bankleitzahl (ggf. vom Scheck abschreiben)

Kontonummer

Kontoinhaber

Geldinstitut/Ort

Datum/Unterschrift

Mit optimierter Startup-Sequence zur schnellen Festplatte

Wem die Diskettenkapazitäten nicht mehr ausreichen, wird sich nach einer Festplatte sehnen. Nun ist eine Harddisk mehr als eine in der Speichermenge erweiterte Diskette. Sehr viele Besonderheiten sind zu beachten, dafür eröffnen sich dem Anwender neue Möglichkeiten. An einer beispielhaften Startup-Sequence werden einige praktische Anwendungen, die auch die Möglichkeiten des FastFilingSystems berücksichtigen, demonstriert.

Nach dem Einbau einer Festplatte beginnt die eigentliche Installation mit der Formatierung. Da eine Festplatte in kleine Häppchen eingeteilt werden kann, sollte man sich Stift und Papier nehmen und die Einteilung — Partionierung genannt — des physikalischen Gerätes „Festplatte“ in mehrere logische Teile skizzieren. Dies ist beim Amiga Betriebssystem 1.2 unbedingt nötig, falls die Plattengröße 51 MByte übersteigen sollte. Aber schon bei 20 MB machen zwei Portionen Sinn. Dabei ist der erste Teil, der als DH0 bezeichnet wird, möglichst klein zu wählen: Auf diesem werden die System-Dateien angelegt. Sie entspricht einer großen Workbench. 4 bis 5 MB sind hier völlig ausreichend. Die Größe wird übrigens in der Mountlist der Boot-Diskette eingegeben. Und zwar durch die Angabe der obersten und untersten Spur dieses Bereiches.
Die richtige Anzahl an Spuren kann nach folgendem Schema ermittelt werden (die entsprechenden Daten sind dabei dem Handbuch zur Festplatte zu entnehmen):
Gewünschte Größe in Bytes/(Anzahl Sektoren * 512 * Anzahl der Köpfe)

So besteht eine 5-MByte-Partition auf einer normalen 20-MByte-Platte aus den Spuren 0 bis 142.
Die zweite Partition DH1 könnte dann mit Spur 143 beginnen und in diesem Beispiel bis zur letzten Spur (613) führen. Das ist eine für diese Größe übliche Einteilung. Auf der großen Partition werden dann sämtliche Daten und weitere Programme abgelegt. In der Mountlist sind für DH1 die Spuren 143 bis 613 voreingestellt.
Bei größeren Festplatten sind bis zu fünf Partitionen angebracht. Die erste bleibt — wie gehabt — die Systempartition. Auf der zweiten — DH1 — kann man wichtige, häufig benutzte Programme und die Datenfiles aller benutzten Software unterbringen. Sicherheitskopien vom Festplatteninhalt sind auf diese Weise meistens nur von dieser Partition notwendig, da die anderen Teile seltener geändert werden. Viele kommerzielle Programme wie DPaint 2 greifen daher per Gadget auf DH1 zu.
Die größte Partition sollte DH2 darstellen — hier sollten alle anderen Anwenderprogramme stehen. Da diese Software noch auf Diskette besteht, muß hier keine Sicherheitskopie angelegt werden.

NoFFS-Teil als Zwischenspeicher für Problemdaten

Auf DH3 befindet sich eine sehr kleine NoFFS-Partition. Dieser Teil sollte nicht mit dem FastFileSystem des 1.3-Betriebssystems formatiert werden, damit eventuell entstehende Probleme an einer Nicht-FSS-Festplatte ausprobiert werden können. Ansonsten wird dieser — langsamere Teil — oft für Zwischenspeicheraktionen benutzt, wenn man sich über die Verwendung einer Datei noch nicht so ganz im klaren ist. Haben Sie übrigens die entsprechenden Eintragun-

gen in der Mountlist für FFS vorbereitet kann dieser Teil durch Angabe des Zusatzes „ffs“ nach dem Format-Befehl auch entsprechend formatiert werden. Um aus einer solchen Partition wieder eine nach dem alten Format zu machen, muß „noffs“ zusätzlich in die Format-Zeile eingegeben werden. DH4 ist etwas ganz Besonderes. Hier wird offensichtlich ein Teil der Platte verschenkt, sind doch statt der vorhandenen sechs Köpfe nur zwei eingetragen. Das kann man sich jedoch bei 65- oder 70-MByte-Platten leisten. Diese Partition ist genauso angelegt, wie eine Diskette! So kann mit dem neuen Diskcopy-Befehl der WB 1.3 eine Diskette direkt auf die Festplatte kopiert werden! Daher sollte auch dieser Teil nicht mit FFS formatiert werden, da das trackdisk.device im ROM dieses Filesystem für Disketten — und nichts anderes ist die Partition nach einem Diskcopy schließlich — noch nicht erkennt. Zu beachten ist allerdings noch, daß die erste Partition — DH0 — nicht mit FFS formatiert werden darf, wenn Sie den 2090A-Autoboot-Controller von Commmodore besitzen.

```
/*System Partition*/
DH0:        Device = harddisk.device   /* Name des Device */
            FileSystem = l:FastFileSystem/* Für FFS nötig */
            Unit = 1                         /* Anzahl Laufwerke (immer 1) */
            Flags = 0                        /* ohne Bedeutung */
            Surfaces = 6                     /* Anzahl Köpfe */
            BlocksPerTrack = 26              /* Sektoren pro Spur */
            Reserved = 2                     /* Reservierte Sektoren (Bootsektor) */
            Interleave = 0                   /* keine Übersetzung durch das DOS */
            LowCyl = 1                       /* niedrigste Spur */
            HighCyl = 51                     /* höchste Spur */
            Buffers = 20                     /* nicht kleiner wählen */
            GlobVec = -1                           /* Für FFS nötig */
            BufMemType = 1                   /* nutzt Chip- oder Fast-RAM */
            Mount = 1                      /* Automatisches einladen des Treibers*/
            StackSize = 1000
#
/*Daten Partition*/
DH1:        Device = harddisk.device   /* Name des Device */
            FileSystem = l:FastFileSystem/* Für FFS nötig */
            Unit = 1                         /* Anzahl Laufwerke (immer 1) */
            Flags = 0                        /* ohne Bedeutung */
            Surfaces = 6                     /* Anzahl Köpfe */
            BlocksPerTrack = 26              /* Sektoren pro Spur */
            Reserved = 2                     /* Reservierte Sektoren (Bootsektor) */
            Interleave = 0                   /* keine Übersetzung durch das DOS */
            LowCyl = 52                      /* niedrigste Spur */
            HighCyl = 220                    /* höchste Spur */
            Buffers = 20                     /* nicht kleiner wählen */
            GlobVec = -1                           /* Für FFS nötig */
            BufMemType = 1                   /* nutzt Chip- oder Fast-RAM */
            Mount = 1                      /* Automatisches einladen des Treibers*/
            StackSize = 1000
#
/*Programm Partition*/
DH2:        Device = harddisk.device   /* Name des Device */
            FileSystem = l:FastFileSystem/* Für FFS nötig */
            Unit = 1                         /* Anzahl Laufwerke (immer 1) */
            Flags = 0                        /* ohne Bedeutung */
            Surfaces = 6                     /* Anzahl Köpfe */
            BlocksPerTrack = 26              /* Sektoren pro Spur */
            Reserved = 2                     /* Reservierte Sektoren (Bootsektor) */
            Interleave = 0                   /* keine Übersetzung durch das DOS */
            LowCyl = 221                     /* niedrigste Spur */
            HighCyl = 734                    /* höchste Spur */
            Buffers = 20                     /* nicht kleiner wählen */
            GlobVec = -1                           /* Für FFS nötig */
            BufMemType = 1                   /* nutzt Chip- oder Fast-RAM */
            Mount = 1                        /* Automatisches einladen des Treibers*/
            StackSize = 1000
#
/*NoFFS Partition*/
DH3:        Device = harddisk.device   /* Name des Device */
            Unit = 1                         /* Anzahl Laufwerke (immer 1) */
            Flags = 0                        /* ohne Bedeutung */
            Surfaces = 6                     /* Anzahl Köpfe */
            BlocksPerTrack = 26              /* Sektoren pro Spur */
            Reserved = 2                     /* Reservierte Sektoren (Bootsektor) */
            Interleave = 0                   /* keine Übersetzung durch das DOS */
            LowCyl = 735                     /* niedrigste Spur */
            HighCyl = 739                    /* höchste Spur */
            Buffers = 5                      /* nicht kleiner wählen */
            BufMemType = 1                   /* nutzt Chip- oder Fast-RAM */
            Mount = 1                       /* Automatisches einladen des Treibe
#
/*Disketten Partition*/
DH4:        Device = harddisk.device   /* Name des Device */
            Unit = 1                         /* Anzahl Laufwerke (immer 1) */
            Flags = 1                        /* ohne Bedeutung */
            Surfaces = 2                     /* Anzahl Köpfe */
            BlocksPerTrack = 11              /* Sektoren pro Spur */
            Reserved = 2                     /* Reservierte Sektoren (Bootsektor)
            Interleave = 0                   /* keine Übersetzung durch das DOS
            LowCyl = 740                     /* niedrigste Spur */
            HighCyl = 819                    /* höchste Spur */
            Buffers = 5                      /* nicht kleiner wählen */
            BufMemType = 1                   /* nutzt Chip- oder Fast-RAM */
            Mount = 1                        /* Automatisches einladen des Treibers
```

Tafel 1: Mountlist zeigt die eingerichteten Partitionen an.

Unter FFS ist nämlich zur Zeit noch kein Autoboot, das heißt ein Booten direkt von Festplatte, möglich! Das Hauptproblem bei der Startup-Sequence ist, daß man, hat man einmal den Spaß daran gefunden, immer mehr Utilities gleich bei Booten installieren möchte. Das führt dann schnell zu einer Startup-Sequence, die für ihre Abarbeitung zwei oder drei Minuten braucht. Daß das nicht so sein muß, soll anhand einiger Beispiele demonstriert werden. Um die Zeit zu verkürzen, sollte die echte Startup-Sequence auf der Diskette natürlich möglichst kurz gehalten werden. Das erste Beispiel sehen Sie in Tafel 3. Hier wird als erstes mit dem 1.3- install-Befehl überprüft, ob auf der Diskette vielleicht ein nicht standesgemäßer Bootblock installiert ist — der meist auf einen Virus hindeutet. So wird sichergestellt, daß auch die Boot-Diskette ohne Schreibschutz bleiben kann, was nötig ist, wenn aus einer Execute-Datei eine andere ausgeführt werden soll, oder wenn Kommandos wie „If“ und „EndIf“ in einer Batch-Datei ausgeführt werden müssen. Doch dazu später mehr. Nun wird sofort die erste — und nur diese — Partition ge„mountet“. Da hier noch alle erforderlichen FFS-Dateien und Treiber von Diskette geladen werden müssen, dauert das natürlich etwas länger. Als nächstes wird das logische C:-Verzeichnis, aus dem sämtliche Befehle geholt werden, auf die Festplattenpartition übertragen. Im Klartext heißt das, das schon der direkt folgende Execute-Befehl von der schnellen Festplatte „geholt“ wird. Dabei wird diese Datei natürlich — wie alle Batch-Dateien — letztendlich im „s:“- Verzeichnis gesucht. Letztere führt dann die eigentliche Festplatten-Startup-Sequence aus, die ich der Einfachheit halber „HDSYS-Startup“ getauft habe. Ist diese abgearbeitet, schließt die echte Startup-Sequence mit einem simplen „endcli >nil:“ das Fenster und die „Boot-Diskette“ kann aus dem Laufwerk genommen werden. Nur falls ein Reset nötig wird, brauchen Sie diese Diskette wieder.

Einmal muß alles kopiert werden

Die große Festplatten-Startup-Sequence ist in Tabelle 4 zu bewundern. Grundsätzlich werden bei allen Aktionen noch mittels „echo“ Texte ausgegeben. Natürlich kann noch eine Menge Zeit eingespart werden, wenn diese Befehle herausgenommen werden. Zuerst werden alle logischen System-Verzeichnisse auf die Harddisk übertragen, was, vor allem dank FFS, sehr schnell vonstatten geht. Nun werden die restlichen Partitionen ge„mountet“. Einleuchtenderweise muß natürlich auch im letzten devs:-Verzeichnis, dem auf der Festplatte, dieselbe Mountlist stehen, wie auf der Boot-Diskette, da ja nun alle Dateien von der Festplatte geholt werden. Letzteres ist auch der Grund, warum der Mount-Vorgang nun wesentlich schneller vorüber ist, als beim Mount von DH0.

Weiter geht's mit dem eventuellen Einbinden der 1.3-Kommandos zur Korrektur bei auftretendes Fehlern mit einem 68010/020-System. Besitzen Sie keinen solchen Prozessor, braucht das Programm auch nicht gestartet zu werden. Des-

Fortsetzung auf Seite 72 ▶

DPaint-Maske für 1000er

Daß die Tastenschablone für DPaint in der AmigaWelt 1/89 für den 2000er und 500er ausgelegt war, frustete die 1000er-Besitzer. Hiermit liefern wir ihre Version nach. Sie enthält außerdem noch eine Korrektur des Handbuchfehlers „Pinsel vergrößern" (auf „Akzent" statt auf „ß") und die nicht dokumentierte Funktion „Farbwechsel" („"-Taste). Die Flight II-Schablone in diesem Heft ist ebenfalls in einer 1000er-Variante abgedruckt.*

> < **_fehlen auf dem 1000er Keyboard unter deutscher Belegung. Sie sind über die Kombination SHIFT,ALT und dem amerikanischen Äquivalent zu erreichen._**

sition
ppeln

n Zentriert Bereich unter dem Cursor
p Paletten-Dialog
q: Kurven
r: R: Rechteck (un-)gefüllt
s: gepunkt. Freihandw.
S Seite zeigen

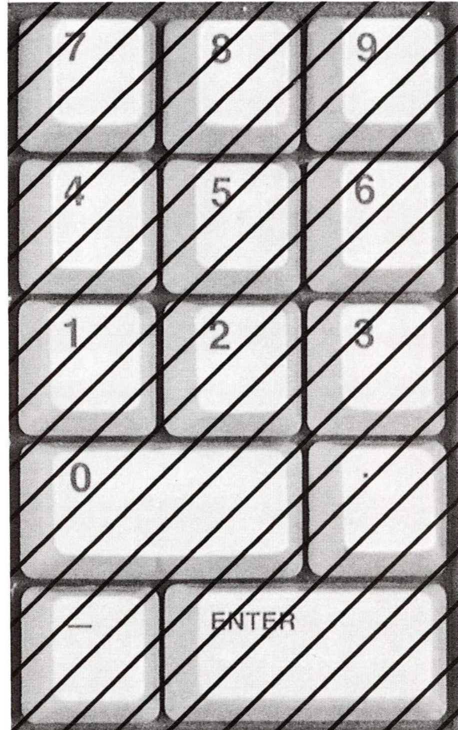
t: Text
, Pick-Werkzeug
Ü, + Farbdurchlauf im gewählten Cycle-Bereich
ß Pinsel verkleinern
` (Akzent) Pinsel vergrößern

u Undo **v** Gerade Linie **x** Flip horizontal
y 90° Drehung **z** Flip vertikal **Y** Strecken
: Zoom kleiner
; Zoom größer
. 1-Pixel-Pinsel, gepunktete Linie
***** Farbwechsel

Schatten — Vermischen — Cycle — Curserpfeil ein/aus — Menüleiste ein/aus — Steuerpult und Menüleiste ein/aus

Ziffernblock/Perspektiven-Modus

7/8	Drehung um die x-Achse
SHIFT 7/8	90°-Drehung x-Achse
9	Rücksetzung x-Achse auf 0
4/5	Drehung um die y-Achse
SHIFT 4/5	90°-Drehung y-Achse
6	Rücksetzen y-Achse auf 0
1/2	Drehung um die z-Achse
SHIFT 1/2	90°-Drehung z-Achse
3	Rücksetzung z-Achse auf Null
0	Rücksetzen aller Achsen auf Null
SHIFT 9	x-Achse fixieren
SHIFT 6	y-Achse fixieren
SHIFT 3	z-Achse fixieren



Abbruch des aktuellen Befehls

Alt-Open Amiga (rechte Seite): Rechte Maustaste

Cursertasten : Rollfunktion (außer Textmodus)

ENTER: Perspektiv-Modus einschalten
- : Bildschirm mit aktuellem Pinsel und aktueller Perspektive füllen
. : Zentrum rücksetzten

el entlang der fixen Achse in perpendikularer Ebene fahren, mit **SHIFT** in größeren Schritten, < und >: Betrachter-Distanz modifizieren

Fortsetzung von Seite 69

```
1> info

Mounted disks:
Unit       Size    Used    Free  Full Errs   Status    Name
VD0:       1.5M       5    3065    0%   0  Read/Write ASDG-RAM
DH4:       880K       9    1749    0%   0  Read/Write HDFloppy
DH3:       390K     107     671   13%   0  Read/Write HDNoFFS
DH2:        40M   22896   57286   28%   0  Read/Write HD
DH1:        13M    7709   18653   29%   0  Read/Write HDData
DH0:       3.9M    5071    2883   63%   0  Read/Write HDSYS
DF0:       880K     315    1443   17%   0  Read/Write WBHD
RAM:         1K       3       0  100%   0  Read/Write RAM DISK

Volumes available:
WBHD [Mounted]
ASDG-RAM [Mounted]
HDFloppy [Mounted]
HDNoFFS [Mounted]
HD [Mounted]
HDData [Mounted]
HDSYS [Mounted]
```

Tafel 2: Der Info-Befehl nach Anmeldung der Partitionen gibt detaillierte Auskünfte.

halb wird diese Zeile, durch das davorstehende Semikolon, als Kommentar behandelt. Nun wird der Systemfont mittels eines Public-Domain-Programmes geändert. Nach dem Setzen der Zeit und der Begrüßung wird die bekannte VD0-RAM-Disk ge„mountet“. Dies ist der einzige theoretische Schwachpunkt der Startup-Sequence. Normalerweise sollte dieses Mount bei möglichst gering fragmentiertem Speicher geschehen, also am besten ganz am Anfang der tatsächlichen Startup-Sequence. Dies kostet jedoch Zeit, und meiner Erfahrung nach gelingt es in den meisten Fällen auch hier noch, die RAM-Disk zu retten. Möchten Sie jedoch in diesem Punkt sicher gehen, sollten die folgenden Zeilen an die erwähnten Stellen kopiert werden. Die Zeile mit dem Copy-Befehl ist ein Trick, um dem Icon der RAM-Disk auf der Workbench ein eigenes Aussehen zu geben. Dazu ist ein Icon vom Typ „Disk“ unter dem Namen ASDG-RAM.info auf die Harddisk zu speichern. Durch die Umbenennung während des Kopiervorganges in „Disk.info“ erhält die Datei ihre eigentliche Bedeutung — das Aussehen des Disk-Icons dieses Gerätes zu bestimmen. Sie dürfen nur nicht versuchen auch in dem Eintrag der Mountlist für die VD0-RAM-Disk „Mount = 1“ einzutragen.

Wartezeit muß exakt eingeplant werden

Dann wird nämlich schon der eigentliche Treiber geladen und der schaut zuerst nach, ob ein „Disk.info“ schon da ist — falls nicht, wird eines generiert — und eben dieses Standard-Icon ist ja nicht das, was gewünscht ist. Weiter geht's mit dem Einladen der Workbench. Der Paramter „-debug“ dürfte inzwischen wohl auch schon hinreichend bekannt sein. Er stellt ein zusätzliches Menü im Titel der Workbench zur Verfügung, von dem aus sämtliche Bibliotheken gelöscht oder der Debug-Modus „betreten“ werden kann. Das „wait“-Kommando muß zwar an dieser Stelle nicht sein, da aber die Workbench automatisch als eigenständiger Task gestartet wird und damit schon weiterläuft, erschien es sicherer, vor der Installation eines Programmes wie „Gomf“, das sehr tief in das System eingreift, alle anderen Aktivitäten erst zu beenden. Überhaupt sollten Sie mit „wait“s nicht zu verschwenderisch, aber auch nicht zu sparsam umgehen. Funktioniert der Start eines Programmes, von der Startup-Sequence aus, aus wirklich nicht erklärbaren Gründen manchmal nicht, schafft der obige Befehl eventuell Abhilfe.

Das zum Umfang der Workbench 1.3 gehörende Programm „FastFonts“ wird nun mit einem entsprechendem Aufruf gestartet. Auch die Installation eines eigenen Tastaturtreibers ist eine bekannte Angelegenheit. Die folgende Zeile sollten vor allem A500-Besitzer unbedingt drin lassen — hier wird sie mittels eines Kommentarzeichens einfach übergangen. Damit wird der langsame, interne Speicher an das Ende der Speicherliste gelegt und damit zuletzt verwendet.

Einige Programme benötigen dies, wenn sie vernünftig arbeiten sollen. Nun werden endlich noch vorhandene, weitere Geräte eingebunden.

Die nächsten Zeilen wurden direkt aus der Startup-Sequence der Workbench 1.3 übernommen und sind nötig, damit Befehle wie „GetEnv“ und „SetEnv“ einwandfrei funktionieren. Der Pfad-Befehl sollte hier nicht als Kommentar stehen, falls mit Workbench 1.2 gearbeitet wird. Unter 1.3 arbeitet man jedoch üblicherweise mit dem Shell und nicht mehr mit dem CLI, so daß diese Definition auch in der Shell-Startup erfolgen kann. Die „Assigns“ für „system:“, „util:“ und „to:“ leiten nur eine Vereinfachung der späteren Arbeit mit dem CLI ein. So können die entsprechenden und vielgebrauchten Verzeichnisse mit kurzen Namen angesprochen werden. In den nun folgenden Zeilen werden sämtliche logischen Geräte hergestellt, die für den Ablauf von bestimmten, auf der Festplatte vorhandenen Programmen nötig sind. Diese notwendigen „Assigns“ erhalten Sie meistens aus den Handbüchern der entsprechenden Software. Um der Kompatibilität der Info-Dateien von BASIC-Programmen, die meist als Default-Tool „Extras:AmigaBASIC“ angegeben haben — was auf das BASIC-Programm der Extras-Diskette hindeutet —, gerecht zu werden wird hier als logisches Laufwerk „Extras:“ das Verzeichnis spezifiziert, in dem sich der AmigaBASIC-Interpreter befindet. An dieser Stelle werden nun die gewünschten Utilities aus den verschiedenen Verzeichnissen aufgerufen. Es handelt sich dabei größtenteils um Public-Domain-Programme der Fish-

```
install df0: check
echo "Mounting Harddisk(s)..."
mount dh0:
assign C: dh0:c
execute dh0:s/HDSYS-Startup
endcli >nil:
```

Tafel 3: So wenig verbleibt als Startup-Sequence auf der Boot-Disk.

Fortsetzung auf Seite 144 ▶

Computermalschule, Teil 7

Tricks mit Sprenkeln

Von Joel Hagen

Bildschirmgemälde müssen nicht steril und aalglatt aussehen. Dank Airbrush. Mit vielen kleinen Pünktchen macht es den Computerbrush zum universalen Pinsel. Es erscheint so, als sei er mal körnig wie Kohle, mal borstig oder haarig wie ein echter Pinsel.

Im deutschen Handbuch zu DeluxePaint heißt es „Luftpinsel", der Anwender kennt es als „Airbrush", und es ist nichts anderes als eine simulierte Sprühdose. Was dieses Utensil vermag, zeigt der Vergleich zwischen Computergemälde und Ölbild. Abgesehen von der strukturierten Oberfläche eines Ölbildes, die je nach Beleuchtung etwas anders wirkt, erkennt man bei der Kunst in Öl den charakteristischen Strich. Borstenpinsel hinterlassen grobe, gestrichelte Spuren, Haarpinsel färben ebenmäßig und erlauben optimale Kontrolle über die Strichbreite. Vielfältige Ausdrucksmöglichkeiten ergeben sich, jenachdem, ob mit farbtriefendem Pinsel, mit fast trockenen Borsten, schwungvoll mit schräg gehaltenem oder senkrecht aufgetupftem Pinsel gearbeitet wird. All diesen Methoden ist eines gleich: Im Gesamteindruck scheint das Zusammenspiel der Bildelemente organisch und gewachsen. Anders das Computerbild. Es erscheint cool und glatt — zu ebenmäßig. Denn der Brush eines Malprogrammes zieht klare, durchgehende Linien gleichbleibender Breite und Färbung. Das läßt sich jedoch ändern. Man muß nur darauf kommen, daß man Airbrush nicht nur als Füllinstrument, sondern auch zum Ziehen von Linien verwenden kann. So läßt sich die Wirkung jedes Pinselstriches weitgehend beeinflussen.

Die vielseitige Airbrush-Funktion ist in nahezu jedem Malprogramm enthalten. Ich verwende jedoch die Bezeichnungen, die in DeluxePaint benutzt werden.

Grundsätzlich kann man jede Brushgröße auswählen. Übungshalber besprüht man mit recht schnellen Bewegungen einen bestimmten Bildschirmbereich. Natürlich kann man dazu große Brushes benutzen, doch basiert die Ausdruckskraft auf besonnener Auswahl zwischen Brushes, die aus einem einzigen Pixel oder aus einer Gruppe unterschiedlicher Pixel bestehen. Mit diesen Brushes entstehen sanft angefärbte Flächen, wie man sie vom Airbrush kennt.

Dann aktiviert man die Hintergrundfarbe mit dem rechten Mausknopf, was die Maus zum Radiergummi umfunktioniert. Mit dem linken Mausknopf malt man mit der aktuellen Farbe im Vordergrund. Belegt man nun die Mausknöpfe mit zwei verschiedenen Tonstufen derselben Farbe und drückt beide Knöpfe abwechselnd, während man einen Bildbereich bearbeitet, resultieren subtile Farbabstufungen. Glanzlichter und Schatten lassen sich auf diese Art und Weise während des Malens setzen. Mischt man zwei nebeneinanderliegende Farben wie grün und blau und verändert die quantitativen Farbanteile, entsteht der Eindruck, hier seien mehr als zwei Farben verwendet worden (siehe Ausschnitt 1 des rechten Bildes). Die Illusion noch feinerer Farbnuancen wird erreicht, wenn man denselben Farbbereich mit zwei weiteren Farben bearbeitet (2). Diese Methode entspricht der des Pointillismus, einem Ableger des französischen Impressionismus. Seurat und Signac experimentierten und arbeiteten mit diesem Ausdrucksmittel.

Auch der Computerkünstler kann davon profitieren und weit mehr Farben vorgaukeln



Mit Airbrush als Line-Tool entstanden die realistischen Fell- und Hautstrukturen des Werwolfes. Das Bild befindet sich auf der Leserdiskette.

als die 16- oder 32-Farben-Palette bietet. Dieses Pixel-Mixen wird auch als „Dithering" bezeichnet. Es wird bei Digitizing-Software eingesetzt, um bei digitalisierten Bildern eine nahezu realistische Farbgebung zu garantieren. Anhand solcher Bilder kann man effektive Airbrush-Farbmischungen studieren.

Vorzugsweise geht man bei der Untersuchung der Farbzusammensetzung folgendermaßen vor: Man digitalisiert – beispielsweise mit DigiView – ein geeignetes Bild, wobei die Scharfeinstellung zwischen 5 und 10 gesetzt wird (in der Farbeinstellung gibt es auch eine Dithering-Funktion). Die Bildvorlage wird mit 16 oder 32 Farben aufgenommen und in ein Malprogramm geladen. Vergrößert man nun unterschiedliche Bildausschnitte, so erkennt man, daß mit beschränkter Farbpalette der Eindruck von weit mehr Farben vermittelt wird (3).

Bevor man das Gelernte umsetzt, muß man sich speziell mit den Charakteristika der Airbrush-Linien befassen. Man wählt das Tool mit der rechten Maustaste. Der Kreis, der daraufhin am Bildschirm erscheint, wird durch Hin- und Herfahren mit der Maus vergrößert oder verkleinert. Da die meisten Anwender Airbrush als Flächenfüller begreifen, wird häufig ein großflächiges Muster benutzt. Tatsächlich aber entfaltet Airbrush seine hervorragende Zeicheneigenschaften erst, wenn man den Kreis stark verkleinert und einen Brush von nur einem Pixel Größe definiert. Dadurch werden die Striche gröber und wirken ähnlich wie mit Holzkohle gezogene Striche, die mit trockenem Pinsel verwischt werden (4).

Die gewünschte Punktdichte auf einer Linie erreicht man durch Erhöhen oder Verringern der Mausgeschwindigkeit. Wer so im Airbrush-Modus malt, erhält abwechslungsreichere Muster und Strukturen, als es die Computerkünstler im typischen Ein-Farben-Verfahren erreichen.

Ideal zum Üben ist grobes Skizzieren angedeuteter, vereinfachter Umrisse mit großzügigen Strichen. Das entstandene Bildskelett versucht man anschließend mit filigraneren Mustern zu füllen. Dabei sollte die rechte Maustaste zum Radieren und Korrigieren benutzt werden. Versuchsweise skizziert man mit vier nahe beieinanderliegenden Farbtönen des Grau- oder Braunbereiches. Es empfiehlt sich dabei – um nicht die Kontrolle zu verlieren – die Mausgeschwindigkeit in den Preferences auf den Wert 2 oder sogar auf 4 zu setzen.

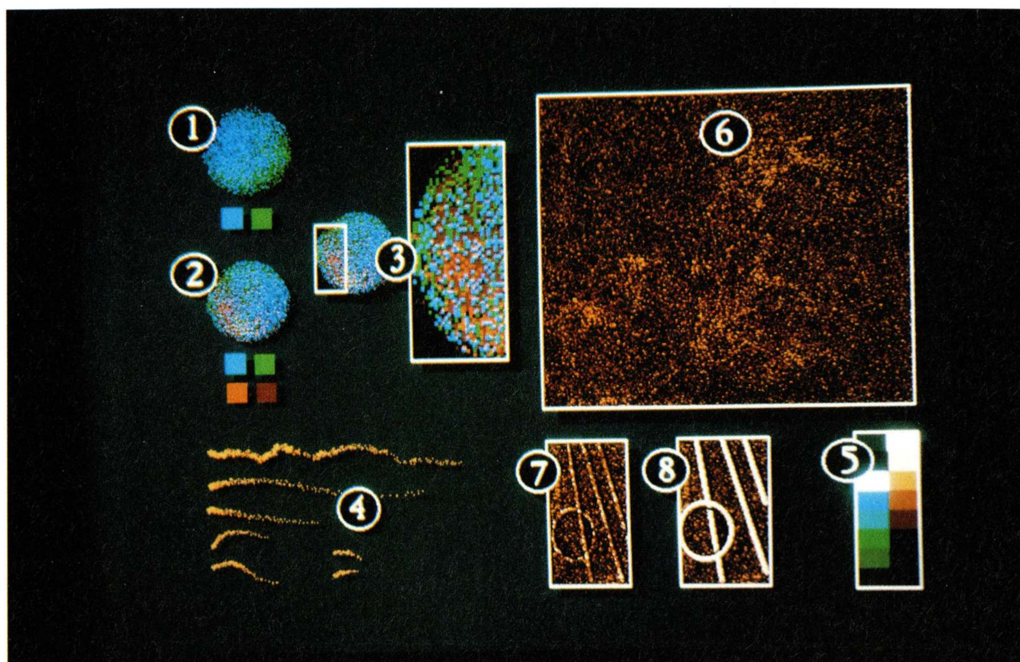
Der Airbrush-Modus bietet berückende Möglichkeiten. Häufig entstehen bei einer Mal-Session ohne Plan und Vorsatz die aufregendsten Bilder. Das Bild entwickelt sich und wächst. In jeder Phase inspiriert es seinen Maler zum Experiment mit neuen Richtungen; ein festes Konzept wird nicht beachtet. Aus den Ideen wird die beste herausgefiltert, durchdacht, verfeinert und ausgearbeitet. Dieses Verfahren intuitiver Annäherung verlangt, ein Bild ohne festes Konzept zu beginnen, es dann aber von Phase zu Phase diszipliniert fertigzustellen.

Genau so ist die Illustration „Inspector", ein Werwolf, entstanden.

Meiner Erfahrung nach ist es ungemein effektvoll, mit einem Farbbereich von fünf bis zehn Abstufungen derselben Grundfarbe zu arbeiten. Das tat ich beim Werwolf. Ich benutzte acht Sepia-Töne (Sepia ist das schwarzbraune Sekret des Tintenfisches) im Bereich von sehr dunkel bis ziemlich hell (5). Diese abgestuften Töne legte ich der Reihenfolge nach in der Palette ab und bestimmte sie als RANGE. Dieser Schritt erlaubte es, Features wie SHADE und BLEND im gewählten Farbbereich zu verwenden, – allerdings in keinem anderen. Die dunkelste Schattierung gab den Hintergrund, und ich färbte so die gesamte Bildschirmfläche ein. Dann ging ich zum mittleren Spektrum über, benutzte den COLOR-Modus und sprühte vorsichtig einen Farbfleck auf den Screen. Den schnitt ich aus und verwendete ihn als Brush, um damit die gesamte Bildschirmfläche schnell und gleichmäßig zu besprenkeln (6). Diese organische Struktur ist viel ansprechender als das sterile Ergebnis des gleichförmigen Gradient Fill. Mit immer kleineren Mustern, in den übrigen Sepia-Tönen meiner Palette hellte ich partiell auf und dunkelte ab. Auf diese Art und Weise schälte ich aussagekräftige Formen und Strukturen aus dem nichtssagenden Flecken. Dann schaltete ich in den SHADE-Modus und aktivierte das Line-Tool. Mit sehr kleinem Brush zog ich hellere und dunklere Linien. Darunter liegende Bereiche dunkelte ich mit der linken Maustaste nach, während ich die rechte zum Aufhellen benutzte. SHADE erzeugt dort feine Linien (7), wo ein massiver Farbstrich klar hervorsticht (8). Das Bild nahm Gestalt an. Mit der Hintergrundfarbe schnitt ich große Bereiche weg und konzentrierte mich auf das Gesicht. Abschließend versah ich Details der Augen, Zähne und Verzierungen im BLEND-Modus mit dezenter Kontrastierung.

Der Werwolf entstand im Hi-Res-Modus mit 640 × 400 Pixeln. Wenn man in der beschriebenen Weise mit Airbrush malt, verschwindet sogar das Interlace-Flimmern. Alternativ kann man die Menü-Farben (0 und 1) auf schwarz und mittelgrau setzen, damit ihren relativen Kontrast verringern und so das Flimmern auch über's Menü beseitigen.

(deutsch von ub)



Trickreiches mit Airbrush: Pixel Ton in Ton, bunt gemischt, dichtgedrängt oder aufgelockert. Die Zahlen im Text beziehen sich auf die Bildbeispiele.

Trickkiste für Programmierer

Nützliche Hinweise für den Umgang mit dem Computer belohnen wir mit mindestens 50 DM pro Kick.

$032 Alert in Basic

Es macht bestimmt keinen schlechten Eindruck, wenn in einem eigenen Basic-Programm nach einem größeren Bedienungsfehler plötzlich eine Alert-Meldung den Benutzer über seine Untat aufklärt. Mit etwas Know-how ist das kein Problem. In der Intuition-Library befindet sich eine Routine namens „DisplayAlert“. Wie ihr Name besagt, läßt sich mit ihr eine solche Alert-Meldung auf dem Bildschirm darstellen. Der aktuelle Screen wird um die Höhe der Box nach unten geschoben, und die Meldung erscheint. Die DisplayAlert-Routine hat folgende drei Parameter:

AlertNumber (= Alarm-Code): Die erste Option werden Sie wahrscheinlich immer auf Null setzen. Sie gibt nämlich an, ob es sich um einen „Dead-End-Alert“ (= nichts geht mehr) oder um einen „Recovery-Alert“ (= wiedergutmachbar) handelt.

StringAdress (= Zeichenkettenadresse): Dieser Parameter bestimmt die Anfangsadresse einer Alert-Message-Struktur (= Alarmmeldung). Es handelt sich also um den Text, der erscheinen soll. Diese Struktur ist folgendermaßen aufgebaut:

```
2 Bytes  X-Koordinate des Textes
1 Byte   Y-Koordinate des Textes
x Bytes  der Text selbst
2 Bytes  Ende oder nicht
```

Wenn die Meldung aus mehreren Einzeltexten besteht (das ist eigentlich immer der Fall), muß man für jeden eine solche Struktur erstellen und an die vorhergehenden anhängen. Das Wort (1 Wort = 2 Bytes) „Ende“ wird nur beim letzten Text auf Null gesetzt, bei allen vorherigen auf 1. In Basic erzeugt man ein Byte mit der CHR$-Funktion, ein Wort mit MIK$. In der Praxis sieht das dann so aus:

```
Meldung$ = MKI$(x) + CHR$(y) + "Text" + MKI$(Ende)
```

Anstelle von x, y und Ende setzen Sie bitte die gewünschten Werte ein. „Text“ ersetzen Sie durch die gewünschte Meldung. Haben Sie mehrere solcher Teilmeldungen vorbereitet, werden sie folgendermaßen zusammengefügt:

```
Gesamtmeldung$ = Teilmeldung1$ + Teilmeldung2$ + ...
```

Height (= Höhe): Mit diesem Parameter können Sie die Höhe der roten Box angeben.

Meistens stehen dem Anwender zwei Möglichkeiten zur Verfügung, auf den Alert zu antworten. Er kann die linke oder die rechte Maustaste betätigen. Falls Sie das in Ihrem Programm abfragen möchten, müssen Sie die Routine als Funktion deklarieren. Das geschieht mit:

```
DECLARE FUNCTION DisplayAlert&() LIBRARY
```

Betätigt der Anwender die linke Maustaste, so ist der zurückgegebene Wert wahr (= −1), andernfalls unwahr (= 0). Bei der Beendigung des Programms muß natürlich mit

```
LIBRARY CLOSE
```

die Intuition-Library wieder geschlossen werden. Alles das finden Sie im folgenden Beispielprogramm wieder. Es ist vielleicht nicht gerade der Sinn eines Alerts, nachzufragen, ob das Programm beendet werden soll, aber zu Verständniszwecken reicht es aus. Normalerweise sollte er nur im Ernstfall erscheinen.

```
 LIBRARY "intuition.library"
 DECLARE FUNCTION DisplayAlert&() LIBRARY
 Zeile1$=MKI$( 45)+CHR$(15)+"Wollen Sie wirklich "+MKI$(1)
 Zeile2$=MKI$(225)+CHR$(17)+"D A S   P R O G R A M M
"+MKI$(1)
 Zeile3$=MKI$(226)+CHR$(17)+"D A S   P R O G R A M M
"+MKI$(1)
 Zeile4$=MKI$(445)+CHR$(15)+"beenden ?"+MKI$(1)
 Zeile5$=MKI$( 45)+CHR$(50)+"Linke Maustaste"+MKI$(1)
 Zeile6$=MKI$(445)+CHR$(50)+"Rechte Maustaste"+MKI$(1)
 Zeile7$=MKI$( 53)+CHR$(58)+"Weitermachen"+MKI$(1)
 Zeile8$=MKI$(492)+CHR$(58)+"Ende"+MKI$(0)

Meldung$=Zeile1$+Zeile2$+Zeile3$+Zeile4$+Zeile5$+Zeile6$+Zei
le7$+Zeile8$
 Schleife:
 INPUT "Bitte 1 Eingeben, 0=Ende ",Eingabe
 IF Eingabe = 0 THEN
   Angeklickt=DisplayAlert&(0,SADD(Meldung$),80)
   IF Angeklickt THEN Schleife
   GOTO Ende
 END IF
 GOTO Schleife
 Ende:
 LIBRARY CLOSE
```

Wenn Sie die Höhe der Alert-Box auf 256 setzen, verschwindet der gesamte darunterliegende Screen. Achten Sie aber vorher darauf, daß genug freier Speicher vorhanden ist, denn sonst kommt es zu einem „echten“ Alert. Der Guru läßt grüßen . . .

(Thilo Riegel)

$033 Preferences einstellen

Um die Preferences einer Diskette zu ändern, ohne daß sich das Programm Preferences auf der zu ändernden Diskette befindet, muß mit dieser Diskette gebootet werden. Während des Bootvorgangs mit der Tastenkombination Ctrl + D Break hervorrufen, dann „DF0:Preferences“ eingeben. Jetzt die Diskette mit dem Programm Preferences ins Laufwerk schieben und erst jetzt Return drücken. Nachdem Preferences geladen ist, die zu ändernde Diskette einlegen, nach Belieben verändern und mit „save“ abspeichern.

(Guido Bär)

$034 DeluxePaint-II-Bug

Die QUIT-Funktion im Picture-Menü von DeluxePaint II (NTSC) hat folgenden Bug: Wählt man sie an, nachdem man am Bild noch etwas verändert hat, so wird gefragt, ob man dieses nicht noch abspeichern wolle. Wer dieses Angebot ausnutzt, wird sich womöglich schon bei der nächsten DeluxePaint-Sitzung gehörig ärgern. Sie werden verdutzt feststellen, daß bei Ihrem Bild einfach die letzte Veränderung fehlt! Der Grund: DPaint ruft mit der QUIT-Funktion aus unerfindlichen Gründen auch noch einmal die UNDO-Funktion auf, bevor das Bild abgespeichert wird. Sie sollten daher Ihre DPaint-Version einmal dahingehend gründlich unter die Lupe nehmen, bevor Sie noch einmal Ihre Kunstwerke über QUIT abspeichern!
(Jörg Schmidt)

$035 Larry ohne Altersnachweis

Um den Altersnachweis bei Leisure Suit Larry zu umgehen, drückt man Alt und X, danach Alt und Z, dann Esc, um dann mit der Eingabe von Return in den Spielgenuß zu kommen.
(Guido Bär)

$036 Joystick in C

Der Joystick wird hauptsächlich für Spiele verwendet. Aber wie programmiert man ihn? Die folgende kleine Funktion liest den Joystick aus und gibt entweder den Wert Null (keine Aktivität am Joystick) oder einen der unten definierten Werte zurück:

```
#define JOY_FIRE   1
#define JOY_RIGHT  2
#define JOY_LEFT   4
#define JOY_UP     8
#define JOY_DOWN  16
USHORT *JOY1DAT = (USHORT *) 0xDFF00C;
UBYTE  *CIAA_PA = (UBYTE  *) 0xBFE001;
BYTE Joystick()
{
 BYTE ret;
 ret=0;
 if(!(*CIAA_PA&128)) ret = JOY_FIRE;
 if(*JOY1DAT & 2) ret |= JOY_RIGHT;
 else if(*JOY1DAT & 512) ret |= JOY_LEFT;
 else if ((*JOY1DAT & 2) ^ (*JOY1DAT & 1)) ret |=
JOY_DOWN;
 else if ((*JOY1DAT & 512) ^ (*JOY1DAT & 256)) ret |=
JOY_UP;
 return (ret);
}
```

(Andreas Schwarzhaupt)

$037 Sprites richtig abschalten

Ist Ihnen auch schon mal bei manchen Programmen (meistens bei Spielen) ein schmaler vertikaler Balken aufgefallen, der sich vom oberen zum unteren Bildschirmrand erstreckt und einfach irgendwo im Bild „klebt"? Nun — Ihr Amiga ist nicht kaputt, verantwortlich sind hier (wie meistens) die schlampigen Programmierer: Sie schalten den Sprite-DMA einfach ab, wobei der Rasterstrahl gerade beim Darstellen eines Sprites (zum Beispiel des Mauszeigers) sein könnte. Wird nun just in diesem Moment das Sprite-DMA-Bit im DMA-Kontrollregister (SDFF096) abgeschaltet, beinhalten die Sprite-Register SPRxDATA und SPRxDATB noch die zuletzt dargestellte Spritezeile. Diese wird nun in jeder Zeile erneut dargestellt, weil der Sprite-DMA keine neuen Sprite-Daten „nachschaufelt": Ein vertikaler Balken entsteht.
Wenn Sie programmieren, können Sie diesen Effekt vermeiden, indem Sie das Abschalten des Sprite-DMA in der vertikalen Austastlücke erledigen, zum Beispiel in Ihrer Copperliste. Eine andere Möglichkeit: Nachdem der Sprite-DMA abgeschaltet ist, können Sie zur Sicherheit der Sprite-Datenregister SPRxDATA und SPRxDATB löschen. Für den Mauszeiger sieht das dann so aus:

```
MOVE.W  #$0020,$DFF096   ;Sprite-DMA aus
CLR.L   $DFF140          ;SPR0DATA & SPR0DATB löschen
```

(Jörg Schmidt)

$038 Problem beim Aufruf von Assemblerroutinen

Ruft man ein Maschinensprache-Unterprogramm (zum Beispiel um Tabellen zu sortieren) beim ersten Mal so auf

```
Anzahl&=n
Tabelle1&=VARPTR(t%(0))
Tabelle2&=VARPTR(t2%(0))
Adresse& =SADD(Maschinenprogramm$)
CALL Adresse&(Tabelle1&,Tabelle2&,Anzahl&)
...
```

kann dies eine tagelange Fehlersuche zufolge haben, denn man sucht den Fehler meist im Maschinenprogramm. Die Ursache liegt aber in der dynamischen Verwaltung von Zeichenketten und Variablen beim AmigaBASIC! Tritt eine Variable zum erstenmal auf, so stellt AmigaBASIC (oder andere BASIC-Varianten) den benötigten Speicherplatz für die Variable zur Verfügung. Dabei kann sich die Lage der bisherigen Variablen ändern. Benutzt man nur die Namen der Variablen, merkt man davon nichts, ermittelt man aber die Adressen per VARPTR-Funktion, sieht man, daß sich die Adressen ändern. Deshalb endet auch obiger Aufruf mit einer Guru-Meditation. Tabelle2& wird neu definiert, und damit ändert sich die Adresse von t%(0) nicht aber der Wert von Tabelle1&, der ja eigentlich die Adresse dem Maschinenprogramm zur Verfügung stellen soll. Jenes sortiert also einen falschen Speicherbereich, was dann meist mit einer Guru-Meditation endet. Deshalb sollte man die benutzten Variablen vorher initialisieren.

```
Tabelle1&=0:Tabelle2&=0:Adresse&=0
```

Anzahl& braucht dabei nicht initialisiert werden, weil die Variablenadressen erst danach mit der VARPTR-Funktion ermittelt werden. Somit ist also sichergestellt, daß die richtigen Adressen übergeben werden.
(Kai Stolze)

Expedition ins AmigaDOS 1.3

Von Muscheln und Befehlszeilen

Unsere Systemforscher profitieren beim weiteren Eindringen in die neue Befehlswelt von AmigaDOS 1.3 von Erfahrungen mit der alten Version. CLI-Spezialisten entdecken in der neuen Shell Eigenschaften des alten Console-Handlers und seinen PD-Erweiterungen.

**Anm. d. Red.: Um auf schnellstem Wege an die für diesen Artikel notwendigen Informationen zu gelangen, wurde hier die Gamma-Version von AmigaDOS 1.3 zugrundegelegt. Unter der Gamma-Version versteht man eine Vorab-Version, die Commodore an Software-Entwickler und bestimmte andere Gruppen herausgibt, damit diese eventuellen Bugs zuleibe rücken. Sie ist die letzte inoffizielle Version, bevor ein Programm in die Ladenregale gelangt. Mittlerweile existieren offizielle Versionen von 1.3 und deutsche Manuals und unseres Wissens sind die Shell-Features unverändert geblieben. Um sicher zu gehen, sollte man nach dem Erwerb der 1.3-Version kurz kontrollieren, ob alle Befehle auf hier beschriebene Weise implementiert sind.*

In AmigaWelt 1/89 wurde ein neues Feature der Workbench 1.3 angesprochen, die Shell. Sie kommt insbesondere passionierten CLI-Benutzern beim Arbeiten sehr entgegen. Hier sollen die verschiedenen Funktionen dieser Shell genauer beleuchtet werden.
Man stellt sich die Shell am besten als aufgemöbeltes CLI vor, mit dem man zunächst einmal wie beim CLI Befehle eingeben und ausführen kann. Darüber hinaus hat die Shell jedoch noch eine stattliche Anzahl neuer Features zu bieten, die eine wesentlich einfachere Handhabung bewirken. Das CLI ist selbstverständlich noch vorhanden — man findet es nach wie vor in der SYSTEM-Schublade auf der Workbench-Diskette. Wenn man jedoch erst einmal einige Zeit mit der neuen Shell gearbeitet hat, möchte man ihre Vorteile gegenüber dem CLI sicher nicht entbehren.

Zum Arbeiten mit der Shell öffnet man die Workbench und klickt das Shell-Icon an. Es sieht dem CLI-Icon sehr ähnlich und könnte durchaus mit ihm verwechselt werden, wenn nicht das Wort „Shell" darunter zu lesen wäre. Das Fenster, das sich nun öffnet, gleicht dem CLI-Fenster wie ein Ei dem anderen. Vorsichtshalber wurde es daher mit „Amiga Shell" beschriftet. Dieses neue Device wird anders gestartet als das CLI: Beim Aufrufen des CLI in AmigaDOS 1.3 wird der File S:CLI-Startup ausgeführt, während bei der neuen Shell S:Shell-Startup für das Starten zuständig ist.

Der Hauptunterschied, den diese Befehle hinsichtlich der Startup-Files aufweisen, ist in erster Linie an den unterschiedlichen Prompts erkennbar, die CLI und Shell am Bildschirm anzeigen. Das CLI-Startup enthält nur einen einzigen, nämlich den traditionellen PROMPT-Befehl

```
PROMPT "%N>"
```

Die Zeichen %N weisen das Prompt an, eine bestimmte CLI-Nummer zu vergeben, wie zum Beispiel beim Standard-Prompt:

```
1>
```

Die Shell-Startup benutzt in ihrem ersten Befehl die Prompt-Anweisung %S :

```
PROMPT "%N.%S>"
```

Die Anweisung %S neben der CLI-Nummer führt dazu, daß das Prompt zusätzlich den Namen des aktuellen Laufwerks umfaßt. Das Standard-Prompt der Shell sieht demnach wie folgt aus:

```
1.SYS:>
```

Man ist jedoch nicht verpflichtet, diese Anweisungen zusammen und in dieser Reihenfolge zu verwenden. Auf Wunsch können weitere, frei wählbare Zeichen in den Prompt-String integriert werden. So könnte man den PROMPT-Befehl in der Shell-Startup wie folgt abändern:

```
PROMPT "Current directory is %S;
 Shell number is %N; enter
 command:"
```

Dies würde die Anzeige des folgenden, eher unbequemen Prompts bewirken:

```
Current directory is SYS:; Shell number is 1; enter
command:
```

Befehlspseudonyme aus eigener Werkstatt

Weiterhin enthält der Shell-Startup-File noch die beiden folgenden Befehle:

```
ALIAS XCOPY COPY [] CLONE
ALIAS ENDSHELL ENDCLI
```

Bei dem ALIAS-Befehl handelt es sich um ein weiteres neues Shell-Feature, mit dessen Hilfe sich „Pseudonyme" für Befehle erzeugen lassen. Natürlich könnte man hierfür auch einfach die Befehls-Files im C Directory der Workbench-Diskette umbenennen, doch würden die fremden Namen einen Gastbenutzer sicherlich verwirren. Mit ALIAS wird ein Befehl jedoch nicht abgeändert, sondern lediglich mit einem Zusatz versehen.

Wie aus den obigen Beispielen hervorgeht, akzeptiert ALIAS zwei Argumente: Einmal den neuen Namen für den Befehl, und weiterhin den Befehls-String, der bei Eingabe des neuen Namens aktiviert werden soll. Der zweite Befehl in unserem obigen Beispiel benutzt dieses Format, um ENDSHELL gleichbedeutend mit ENDCLI zu machen. Hat man einem Befehl einen Alias-Namen gegeben, kann dieser Befehlsname überall dort verwendet werden, wo man auch den Originalbefehlsnamen einsetzt.

Der erste Alias-Befehl in unserem obigen Beispiel veranschaulicht eine äußerst leistungsfähige Alias-Option: Durch Einbinden von [] in den Befehls-String lassen sich Argumente von der Alias-Befehlszeile zum Original-Befehl übertragen. So führt zum Beispiel der String

```
XCOPY FILE1 FILE2 CLONE
```

zum gleichen Ergebnis wie die Eingabe von

```
COPY FILE1 TO FILE2 CLONE
```

Die Shell setzt für [] all die Zeichen ein, die nach dem Befehls-Aliasnamen eingetippt werden.
Es ist zu beachten, daß die Befehls-Strings in diesen Beispielen nicht in Anführungsstriche eingefaßt sind.
Im Gegensatz zu AmigaDOS-Namen, die unbedingt in Anführungsstrichen stehen müssen, sofern sie Leerzeichen enthalten, sollten um diesen Befehls-String keine Anführungszeichen gesetzt werden.

So kann man mit

```
ALIAS CAT "TYPE"
```

den Inhalt eines File mit dem Unix-typischen Befehl

```
CAT FILE1
```

betrachten. Gibt man jedoch statt dessen ein:

```
ALIAS CAT TYPE
```

akzeptiert die Shell zwar das ALIAS, doch gibt man

```
CAT FILE1
```

ein, erscheint die Fehlermeldung:

```
Unknown command "TYPE"
```

Darüber hinaus gibt es noch zwei weitere ALIAS- Optionen. Alle aktuellen Pseudonyme lassen sich auf den Bildschirm rufen, indem man

```
ALIAS
```

eingibt. Zum Löschen eines Pseudonyms tippt man diesen ohne den Befehls-String ein, wie zum Beispiel bei

```
ALIAS CAT
```

Ein Aliasname kann neu definiert werden, indem man einfach einen neuen Befehls-String definiert.
In der Beschreibung des RESIDENT-Befehls in der vorigen Ausgabe wurde erwähnt, daß alle Shells sich ein einziges RESIDENT-Befehls-Set teilen. Dies gilt jedoch nicht für Pseudonyme; hier besitzt jede Shell ihr eigenes Set. So ist es möglich, mehrere Shells gleichzeitig auszuführen, wobei jede auf ein anderes Pseudonym-Set zurückgreift.

Einfache Ausführung auch von Batch-Files

Pseudonyme sind bei CLI-Befehlen vielleicht recht nützlich, doch viele Amiga-User arbeiten mit Batch-Files, die sie ebensooft ausführen wie einzelne Befehle. Ein Batch-File wird mit dem EXECUTE-Befehl über CLI eingeleitet. Mit der Shell läßt sich ein Batch-File auf direkte Weise ausführen, sofern das Script-Bit (S) eingestellt ist. Das S-Bit ist eines der neuen Schutzbits der Version 1.3. Es wird mit dem Befehl PROTECT eingestellt:

```
PROTECT FILE1 +S
```

Erst wenn das S-Bit eines File eingestellt wurde, kann dieser durch Eingabe des Filenamens in die Shell ausgeführt werden. Der Batchfile FILE1 wird wie folgt gestartet:

```
FILE1
```

Die Shell benutzt zum Ausführen von Files noch immer den EXECUTE-Befehl — dies ist wohl nichts besonderes — doch erspart sie dem User das Auseinanderhalten von Programmen und Batch-Files bei den aktuellen Befehlen. Mit Hilfe des RESIDENT-Befehls in der S:Startup-Sequence wird der EXECUTE-Befehl speicherresident gemacht, mit dem Ergebnis, daß sich Batch-Files schnell und problemlos ausführen lassen. Übrigens ist es äußerst praktisch und empfehlenswert, häufiger benutzte Batch-Files im S: Directory unterzubringen. Mit dem PATH-Befehl in der Startup- ▶

Sequence läßt sich das Directory in den Befehls-Suchpfad integrieren, wodurch das Starten dieser Batch-Files von jedem Directory aus ermöglicht wird.

Eine neue Shell — kein Problem

Genauso wie man mehrere CLIs gleichzeitig bedienen kann, lassen sich auch verschiedene Shells nebeneinander ausführen. Selbst eine Kombination dieser beiden Device-Formen ist zur gleichen Zeit ausführbar.
Leitet man einen neuen Prozeß im CLI mit NEWCLI ein, so wird dies bei der Shell analog dazu mit NEWSHELL bewerkstelligt. Die mit diesen Befehlen geöffneten Shells und CLIs arbeiten mit den standardmäßigen CLI- und Shell-Windows der AmigaDOS-Version 1.3. Sowohl NEWCLI als auch NEWSHELL verfügt über eine Option, mit der sich Lage, Größe sowie der Name (title) des Fensters bestimmen lassen. Zu diesem Zweck werden dem AmigaDOS-Window-Handler besondere Angaben mitgeteilt. In der 1.2-Version von AmigaDOS existiert lediglich ein einziger Window-Handler, nämlich CON:. Als Option zu NEWCLI konnte er auf folgende Weise eingesetzt werden:

```
NEWCLI CON:x/y/Breite/Höhe/Name
```

X und Y geben die Anfangszeile beziehungsweise -spalte wieder, und über die Breite und Höhe wird die Größe des Fensters festgelegt. Der Name erscheint im oberen Fensterbalken.
So wird mit dem Befehl

```
NEWCLI CON:20/10/400/200/MeinCLI
```

ein neues CLI mit einem 400 x 200 Pixel großem Fenster angelegt, das in Pixelspalte 30 und in Pixelzeile 10 beginnt und mit „MeinCLI" bezeichnet ist.
Möchte man ein neues Shell-Window erzeugen, geht man auf ähnliche Weise vor, wobei man hier jedoch zusätzlich auf den neuen Befehls-Handler NEWCON: zurückgreifen kann. Abgesehen davon sind die Optionen mit denen des CLI identisch.
Durch Eingabe von

```
NEWSHELL NEWCON:20/10/400/200/MeinCLI
```

wird ein neues Shell-Window an der gleichen Position wie das obige CLI-Fenster erzeugt. Beide Befehle arbeiten stets mit ihrem eigenen Window-Handler, auch wenn die neue Window-Information nicht angegeben wird. In diesem Fall legt die Shell die standardmäßige Window-Information zugrunde.

Bei Verwendung von NEWSHELL wird zweierlei vorausgesetzt: Einmal daß sich L:Shell-Seg, der Code, der die Shell implementiert, bereits resident im Speicher befindet, und zum zweiten, daß der NEWCON: Window-Handler bereits installiert wurde. Im Gegensatz zum CON: Handler, der nicht zu installieren ist, besitzt der NEWCON: Handler einen Eintrag in der DEVS:MountList, der vor dem Gebrauch installiert werden muß. Sicherheitshalber enthält die Standard-Startup-Sequence die folgenden beiden Befehle, die dafür sorgen, daß diese Voraussetzungen erfüllt werden:

```
RESIDENT CLI L:SHELL-SEG
SYSTEM PURE
MOUNT NEWCON:
```

Zwischen den NEWCON: und den CON: Window-Handlers besteht ein weiterer Unterschied, der sich bemerkbar macht, wenn man mit dem Befehl TYPE Files, zum Beispiel Texte aus Textverarbeitungen, die Steuerzeichen enthalten, am Bildschirm anzeigt. Im Gegensatz zu CON: gibt NEWCON: nämlich keine Steuerzeichen am Bildschirm aus. Diese Zeichen werden allerdings nicht aus den Files entfernt; es erscheinen anstelle der Steuerzeichen einfach Leerzeichen.

Erleben der Befehlsgeschichte

Man könnte nun denken, daß es eigentlich egal ist, ob ein neuer Window-Handler eingebaut wurde oder nicht. Sicherlich wieder nur ein Bonbon für Programmierer. Oder?
Nicht ganz, denn diese Eigenschaft gehört zu den besten Features, die die neue Shell aufzuweisen hat, da sie eine große Vielfalt an Optionen zu bieten hat.
Hat man sich, des Tippens überdrüssig, nicht schon einmal gewünscht, einen zuvor eingegebenen Befehl zu neuem Leben zu erwecken? Mit der neuen Shell ist es möglich, durch die zuvor eingetippten Befehle zurückgehen, dank eines Feature, das Commodore mit „History Buffer" bezeichnet. Hierbei handelt es sich um einen Kreis-Buffer, der die letzten 2 KByte eingegebener Befehle speichert.

```
CD S:
DIR
TYPE CLI-STARTUP
TYPE SHELL-STARTUP
LIST
```

Möchte man nun TYPE CLI-STARTUP erneut aktivieren, könnte man, statt den Befehl nochmals einzugeben, den im History Buffer gespeicherten und bereits ausgeführten Befehl erneut aufrufen. Hierzu betätigt man einfach einmal die Taste PFEIL OBEN. Bei nochmaligem Drücken begibt man sich mit dem Cursor auf den Befehl darüber und so weiter, bis man sich durch die gesamten 2 KByte Buffer gearbeitet hat und zur leeren Befehlszeile zurückgelangt.
Drückt man in unserem Beispiel zweimal PFEIL OBEN, würde der Cursor auf folgende Zeile positioniert:

```
TYPE CLI-STARTUP
```

Nach dem Drücken von Return wird der Befehl ausgeführt, geradeso, als wenn man ihn eingetippt hätte.
Durch Betätigung der PFEIL UNTEN-Taste begibt man sich im History Buffer auf den nächstfolgenden Befehl. Ist man also versehentlich eine Zeile zu hoch geraten, kehrt man mit PFEIL UNTEN wieder an die gewünschte Stelle zurück. Befindet sich der Cursor auf der leeren Befehlszeile und die PFEIL UNTEN-Taste wird gedrückt, passiert natürlich gar nichts, weil der History Buffer leer ist.
Es ist ebenfalls möglich, den History Buffer auf einen bestimmten Befehl hin zu durchsuchen. Erinnert man sich nicht mehr an den genauen Namen des CLI-Startup-File, kann man folgendes eingeben:

```
TYPE C
```

Anschließend drückt man SHIFT <Pfeil oben>. Nun sucht die Shell den Befehls-Buffer rückwärts durch, solange, bis sie auf den ersten Befehl stößt, der mit dem eingegebenen Textstring übereinstimmt. Findet sie kein „Match", wird die Befehlszeile nicht verändert. Andernfalls erscheint der gesuchte Befehl vollständig ausgeschrieben in der Befehlszeile:

```
TYPE CLI-STARTUP
```

Zum Durchsuchen des History Buffers nach rückwärts kann man statt SHIFT <Pfeil oben> auch CTRL-R drücken.

Ist man im History Buffer bereits weiter zurückgegangen, besteht die Möglichkeit, mit SHIFT <Pfeil unten> oder mit CTRL-B an das Ende, also an die leere Befehlszeile vorzuzuspringen.

Geschichte läßt sich verändern

Durch das Re-aktivieren bereits eingegebener Befehle läßt sich sehr viel Zeit sparen. Gelegentlich möchte man jedoch einen „alten" Befehl nicht in seiner ursprünglichen, sondern in abgeänderter Form verwenden. Auch hierbei ist einem die Shell behilflich.

Zunächst einmal überträgt man auf oben beschriebene Weise den alten Befehl aus dem History Buffer in die aktive Zeile. Dann editiert man diesen Befehl. Jawohl, er wird editiert. In der Shell lassen sich Befehle durchaus abändern, sowohl die gerade eingegebenen als auch die Befehle aus dem History Buffer.

In weniger großzügigem Maße lassen sich natürlich auch Befehle im CLI editieren. So kann man mit BACKSPACE das Zeichen links vom Cursor entfernen und mit CTRL-X die gesamte Zeile löschen. Bei der Shell kommen allerdings noch einige Editiertasten hinzu.

<Pfeil links> und <Pfeil rechts> gestatten das Versetzen des Cursors in der Befehlszeile. Nach Drücken von CTRL-W springt der Cursor bis zum nächsten TAB-Stop, und mit SHIFT <Pfeil links> oder CTRL-A wird er an den Anfang beziehungsweise mit SHIFT <Pfeil rechts> oder CTRL-Z an das Ende der Befehlszeile positioniert.

Gibt man an der neuen Cursorposition Text ein, wird der vorhandene Text nach rechts verschoben.

Selbstverständlich lassen sich ungewünschte Zeichen auch entfernen. Mit BACKSPACE wird das Zeichen links vom Cursor gelöscht, während DEL das Zeichen an der aktuellen Cursorposition zum Verschwinden bringt.

Textblöcke sind auf verschiedene Arten löschbar. CTRL-X führt dazu, daß die gesamte Zeile geleert wird. CTRL-U löscht die Zeichen vom Zeilenanfang bis hin zur Cursorposition, wobei das Zeichen an der Cursorposition an den Zeilenanfang plaziert wird. Die Betätigung von CTRL-K bewirkt das Gegenteil: Alle Zeichen von der Cursorstelle bis zum Ende der Zeile werden gelöscht, und hier wird — im Gegensatz zu CTRL-U — darüber hinaus noch das Zeichen an der Cursorposition entfernt.

Aufgrund seiner Fähigkeit, bereits ausgeführte Befehle erneut zu aktivieren und zu editieren, sowie auch in Anbetracht seines leistungsstarken Funktionspaketes, wird mit der neuen Shell ein vereinfachtes und schnelleres CLI geboten. Ist man erst einmal mit den Raffinessen der 1.3-Version vertraut, macht das Arbeiten sicherlich noch mehr Spaß.

(M. L. v. Name/B. Catching/S. Bröker)

Digital gegen original

Digitalisieren ist im Prinzip nicht schwierig. Doch läßt meist die Qualität des Ergebnisses zu wünschen übrig. Tips vom Fachmann helfen.

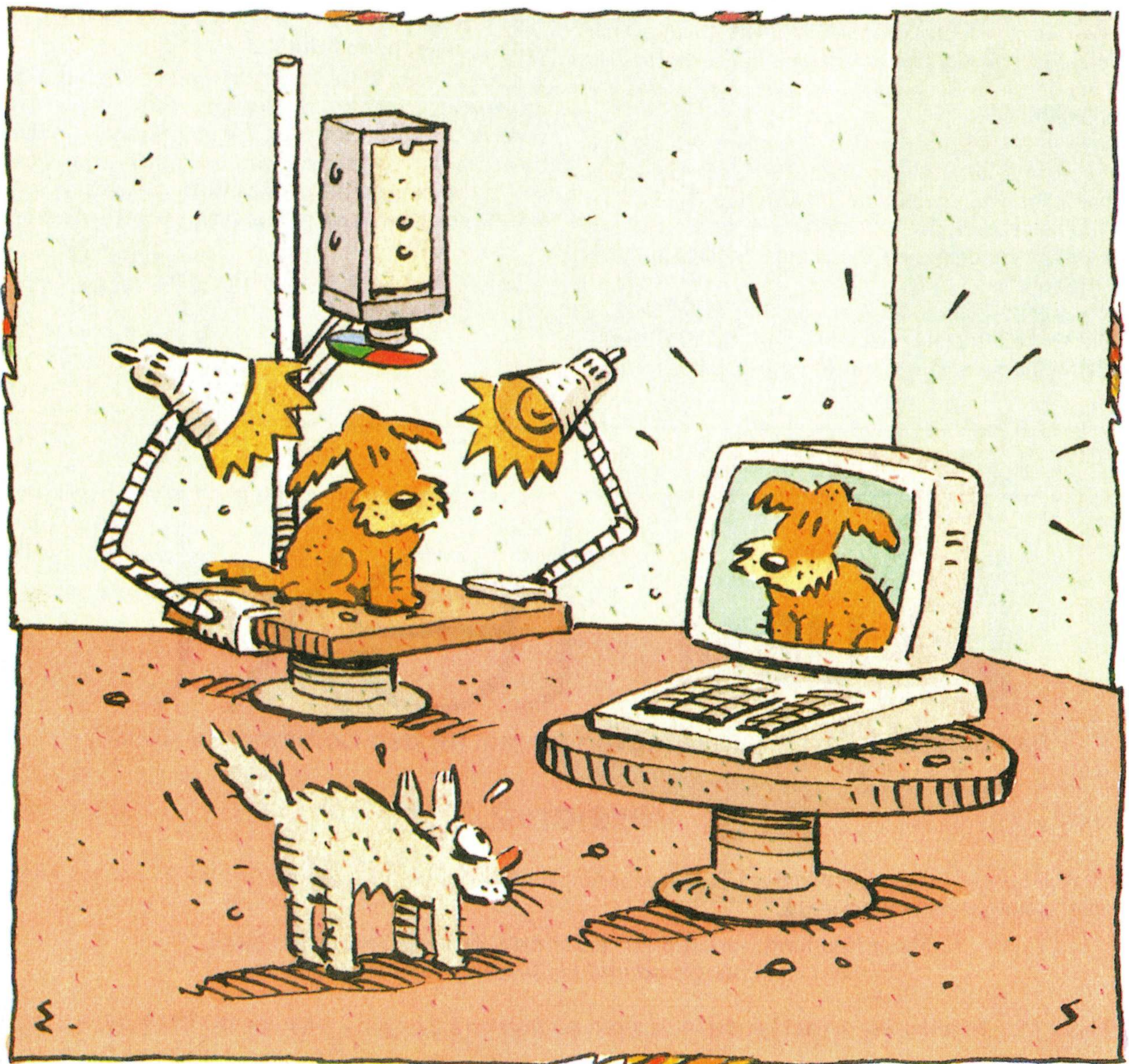


Illustration: Elwood H. Smith

Kamera auf das Motiv richten, Farbscheibe drehen, Digitalisier-Software aktivieren, — damit ist der Vorgang des Digitalisierens bereits umrissen. Doch soll das Digi-Bild seiner Vorlage auch wirklich ähnlich sehen, muß man die geeigneten Voraussetzungen schaffen und wichtige Kleinigkeiten beachten. Bei hochwertigen Amiga-Digitizern entstehen kaum Probleme. Mit diesen kostspieligen Geräten kann man verblüffend realistische Bilder auf den Bildschrim bringen. Mängel bei Durchschnittssystemen lassen sich jedoch teilweise kompensieren.

Für die angeführten Beispiele wurden verwendet: Digi-View von NewTek, der momentan gängigste Digitizer, ein Testrite Copy-Stand sowie eine Panasonic WV-1410-Videokamera.

Unerläßlich für optimale Digi-Bilder ist die angemessene Beleuchtung. Wenn das Licht nicht stimmt, kann das Ergebnis nicht zufriedenstellend sein. Zur Überprüfung der Lichtverhältnisse digitalisiert man zunächst eine homogene mittelgraue Fläche. Auf diese Weise läßt sich erkennen, welche Stellen ungleichmäßig beleuchtet sind. Bei leichten Diskrepanzen behilft man sich mit weißem Karton, der um den Sockel des Copy-Stands gelegt wird. Die weiße Pappe reflektiert das Licht und leuchtet dunklere Stellen aus. Ärgere Lichtprobleme löst man mit einem Diffusionsfilter oder nichtbrennbarem Kunststoff. Oder man bedeckt den Reflektor mit einem Stück Stoff. Diffusionsfilter dämpfen das Licht, so daß die Fläche gleichmäßiger angestrahlt wird. Da diese Maßnahme die Lichtstärke verringert, ist die Leistung der Strahler um zirka 25 Watt zu erhöhen.

NewTek gibt an, daß Digi-View bei Licht von einer Farbtemperatur zwischen 2800 Kelvin (durchschnittliche Farbtemperatur einer Innenraumbeleuchtung) und 6000 Kelvin (durchschnittliche Farbtemperatur des Tageslichts) angewendet werden kann. Diese Angabe läßt sich überprüfen, indem man bei verschiedenen Lichtstärken digitalisiert. Der Wertevergleich zeigt, daß selbst professionelle Fotostrahler zu hell für Kamera und Filter sind. Am natürlichsten erscheint die Farbe bei der Verwendung weißer Leuchtstofflampen, deren Temperatur etwa bei 4500 Kelvin liegt. Das Licht verteilt sich gleichmäßig, die Klimaanlage kann meist ausgeschaltet bleiben.

Bei intensiver Lichteinstrahlung wird das Papier warm und wellt sich. Ein Fokussieren der Kamera ist kaum möglich. Der Profi greift in diesem Fall zur Vakuumstaffelei. Im oberen, flachen Teil der Staffelei saugt eine Vakuumpumpe durch viele kleine Öffnungen Luft an.

Staubsauger halten Bilder flach

Legt man die Vorlage über diese Öffnungen, wird sie durch den Sog flach gehalten. Eine Staffelei ohne Vakuumpumpe läßt sich notfalls mit einem Staubsauger improvisieren. Lediglich Bücher bereiten beim Vakuumtisch Probleme, es sei denn, man reißt die Seiten heraus.

Bild oder Buch können zur Not mit einer Glasplatte beschwert werden. Beim Einstellen der Beleuchtung ist dann jedoch äußerst behutsam vorzugehen, um eine partielle Überhitzung zu vermeiden. Gegen deren Ursache — starke Spiegelungen — helfen polarisierende Filter auf Kamera und Lichtquellen. Normales Fensterglas gibt allen Farben einen Grünstich. Mit den RGB-Schiebern im Farbmenü oder durch Auswechseln der RGB-Filter läßt sich diese Farbverfälschung aufheben.

Selbst bei optimaler Beleuchtung und flach anliegender Vorlage kommt es doch zu ganz erheblichen Abweichungen vom Original. Denn Amiga schafft es nicht, wirklich alle Farben und Schattierungen eines Voll- oder Halbtonbildes aufzulösen. Wie erfolgversprechend ein Motiv ist, läßt sich anhand von Erfahrungswerten feststellen. Bilder mit strukturreicher Oberfläche eignen sich am besten. Größere Flächen mit satten Farben machen Schwierigkeiten bei der Belichtung. Strahlend blauer Himmel und weiße Wände sind problematisch, auch ist von detailreichen Bildern mit starken Kontrasten abzuraten. Gerade das Auflösen von Details in Schattenflächen gilt als besondere Herausforderung; in den meisten Fällen ergeben sich nämlich lediglich schwarze Kleckse. Wer sich trotzdem daran wagt, kann sich eines Kunstgriffs bedienen: die günstigste Belichtung wichtiger Bildelemente wird in jeweils einer Datei abgespeichert. Anschließend vervollkommnet man das Resultat mit einem Mal- oder Bildverarbeitungs-Programm.

Damit Farben von Original und Digi-Bild weitgehend kongruent sind, stimmt man die Systemfarben vor jedem Digitalisiervorgang mit einer Farbvorlage ab. Oder man beschafft sich den Kodak-Farbtrennungs-Ratgeber für Profis. Er enthält eine Grauskalentabelle mit 20 Stufen sowie eine Chip Chart mit hoher Farbpräzision. Beide werden auf ein Stück festen Kartons geklebt und digitalisiert, — und zwar möglichst dann, wenn die Strahler neu sind. Hierbei ist der F-Stop der Kamera, die Position der Strahler und die Farbeinstellung im Programm so zu wählen, daß die Farben am Bildschirm denen der Tabelle nahekommen. Man notiert nun die Einstellungen und klebt sie zur Orientierung auf die Kamera. Sind die zuletzt eingestellten Farben nicht korrekt, müssen sie nochmals justiert werden, wobei wieder von den ursprünglichen Werten auszugehen ist. Im Zweifelsfall legt man die Tabelle neben die Vorlage ins Bild und justiert erneut.

Eine Kamera tut nicht immer, was ihr Benutzer will. Sobald der F-Stop auf der Linse justiert ist, um ein überbelichtetes Bild dunkler zu machen, setzt die Autogain-Vorrichtung die Kamera mitunter in die Ausgangsposition zurück und erzeugt ein sehr blasses Bild. Das läßt sich zwar nicht vollends ausschließen, doch kann man diesen Effekt folgendermaßen mindern: Jene Teile des Bildes, die nicht digitalisiert werden sollen, sind abzudecken, denn große helle oder dunkle Flächen werden vom Autogain stets überkompensiert. Diese Unzulänglichkeit kann man bei extrem über-und unterbelichteten Bildern zu seinem Vorteil nutzen. Je nach Vorlage verhelfen weißer oder schwarzer Karton zum gewünschten Belichtungsgrad.

In jede Digitizer-Utensilbox gehört ferner ein Stück schwarzen Samtes, das weder Lichtreflektionen verursacht noch die Farben beeinflußt. Für Aufnahmen, die ein Objekt in der Schwebe vorgaukeln sollen, ist dieses einfache Hilfsmittel unerläßlich und spart das Wegschneiden des Hintergrundes.

Beim Digitalisieren ist es wichtig, daß die Geräte keinen Vibrationen ausgesetzt sind, da sich sonst Verzerrungen und unerwünschte Linien einschleichen. Um Erschütterungen durch den Drucker und die Bedienung der Tastatur zu vermeiden, stellt man den Copy-Stand auf einen separaten Tisch. Ist der Raum mit Parkettfußboden ausgelegt, sollte man den Tisch in eine erschütterungsfreie Ecke des Raums rücken.

Malprogramme kurieren Farbfehler

Nachdem man die Farbscheibe gedreht hat, wartet man einige Sekunden und zeichnet erst dann einen neuen Farbauszug auf. Dies empfiehlt sich, da das Kameragestell nicht sehr stabil ist und noch eine Weile nachschwingt, nachdem man es berührt hat. Für Zappelphilippe, die partout nicht stillstehen können, gibt es den Digi-Droid von NewTek, der die Farbfilter selbsttätig dreht.

Farblich verunglückte Bilder, die man nicht noch einmal digitalisieren will, rettet man mit DeluxePaint (Electronic Arts) oder PixMate (Progressive Peripherals & Software). Roter, ▶

Bezugsquellen und Preise:

(alle Preise sind Zirka-Preise)
Butcher 2.0 (Eagle Tree) PAL-Version: 80 Mark,
bei: Amiga Oberland, CompuStore

DeluxePaint II PAL (Electronic Arts): 250 Mark,
bei: Philgerma

DeluxePhotoLab (dt.):
260 Mark,
bei: CompuStore, Philgerma

DigiDroid: 160 Mark,
bei: Amiga Oberland, Atlantis, Philgerma

DigiPaint (New Tek) 125 Mark,
bei: DTM, Philgerma

Digitize-Paket mit S/W-Videokamera incl. HiRes-Objektiv und DigiView PAL 3.0:
998 Mark,
bei: Optivision

DigiView (NewTek) PAL:
330 Mark,
bei: Amiga Oberland, CWTG, CompuStore

Deutsche Anleitung zu DigiView 2.0: 30 Mark,
bei: Softwareversand Müller

Frame Capture (Mimetics) NTSC,
bei: Atlantis, Intelligent Memory

Frame Grabber (Progressive),
bei: Atlantis, GTI, Jumbosoft, Soyka

Hama Reprogestell (kann anstelle des Testrite Copystandes verwendet werden),
bei: Fotofachhandel

Megavision-Grafikkarte (Sang Computer) PAL: 2000 Mark,
bei: CompuStore

Perfect Vision (SunRize),
bei: Intelligent Memory

Photon Paint (MicroIllusions) mit dt. Handbuch: 245 Mark,
bei: videoLOFT, CompuStore

ReaSyn (Mimetics) NTSC,
bei: Atlantis, GTI, CompuStore, Intelligent Memory

Sharp Scanner 300:
12 500 Mark,
bei: CompuStore

Adressen:

Amiga Oberland, A. Koppisch, Hohenwaldstr. 26, 6374 Steinbach, Tel. 0 61 71/7 18 46

Atlantis, Dunantstr. 53, 5030 Hürth, Tel. 0 22 33/4 10 81

CompuStore, Fritz-Reuter-Str. 6, 6000 Frankfurt 1, Tel. 0 69/56 73 99

CWTG, Joachim Tiede, Bergstr. 13, 7109 Roigheim, Tel./Btx: 0 62 98/30 98

DTM, Poststr. 25, 6200 Wiesbaden, Tel. 0 61 21/56 00 84

GTI, Zimmersmühlenweg 73, 6370 Oberursel, Tel. 0 61 71/36 25 22

Intelligent Memory, Borsigalle 18, 6000 Frankfurt 60, Tel. 0 69/41 00 71

Jumbosoft, Horemansstr. 2, 8000 München 19, Tel. 0 89/1 23 40 65

Optivision, Aachener Str. 78—80, 5000 Köln 1, Tel. 02 21/56 14 60

Philgerma, Barerstr. 32, 8000 München 40, Tel. 0 89/28 12 28
oder: Baroperstr. 337, 4600 Dortmund 50, Tel. 02 31/75 92 92

Softwareversand Müller, Dorfstr. 1, 8852 Rain-Unterpeiching

Soyka, Hattingerstr. 685, 4630 Bochum 5, Tel. 02 34/4 98 25—26

videoLOFT, Fiedlerstr. 22—32, 3500 Kassel, Tel. 05 61/87 33 99

grüner und blauer Auszug lassen sich bei Verwendung von PixMate, DeluxePhotoLab (EA) oder Butcher (Eagle Tree Software) einzeln als IFF-Dateien speichern. Dann lädt man zwei Aufnahmen in DeluxePaint ein — eine als Bildschirm und eine als Brush — und schiebt beide übereinander. Dann schaltet man auf den freien Bildschirm um und trägt den Brush auf. Im Anschluß daran wird die dritte Aufnahme als Brush geladen und der Prozeß mit den beiden Auszügen wiederholt, deren Farbüberdekkung korrekt ist. In diesen Fällen besteht ebenfalls die Möglichkeit, auf die automatische Merge-Funktion von PixMate zurückzugreifen.

Beim Digitalisieren großer Bilder, die vom Blickfeld der Kamera nicht völlig erfaßt werden, verwendet man einfach ein höheres Kameragestell. Das läßt sich mit einem geeigneten Stahlrohr bewerkstelligen. Aber es gibt eine kostengünstigere Alternative. Man schraubt die Lampen zirka 3-15 cm von den Sockelecken nahe dem Kameragestell an. Den Copy-Stand stellt man so auf, daß die Lampenhalterung und das Stützgestell der Kamera genau über die Tischkante reichen. Dann dreht man den Arm der Kamera auf dem Gestell herum, bis diese auf den Boden zeigt. Wer Standardlampen verwendet, schraubt Kameraarme und Lampen ab und bringt sie umgekehrt an, so daß das Licht auf den Boden gerichtet ist. Großflächige Vorlagen erfordern zusätzliche Lichtquellen, die eine gleichmäßige Beleuchtung gewährleisten.

Hoher Stromverbrauch und Lichtprobleme werden vermieden,und die Qualität der Bilder wird trotzdem erheblich besser, wenn man Negative und Diapositive direkt digitalisiert. Durch das Justieren der mechanischen oder Back-Focus-Vorrichtung läßt sich die grobe Einstellung (für extreme Nahaufnahmen) kostspieliger Linsen emulieren; dadurch läßt sich die Kamera bis zur Linse fokussieren.

Beim Dia-Digitizing mit der Back-Focus-Vorrichtung

schiebt man ein Stück Glas oder matten Kunststoff zwischen Linse und Farbscheibe. An den unteren Rand des Glases wird ein Stück weißes Papier oder matter Kunststoff geklebt, wobei die Farbscheibe nicht berührt werden darf. Nun richtet man die Lichtquelle, die 30-70 cm vom Glas entfernt sein soll, auf die Kamera. Das Dia wird auf das Glas gelegt, Beleuchtung und Kamera werden eingeschaltet. Der Back-Focus wird justiert, bis das Dia scharf eingestellt ist. Dann wird digitalisiert wie gewohnt.

Daß die Standardlinse der Panasonic-Kamera auch bei schwachem Licht für akzeptable Ergebnisse sorgt, liegt daran, daß sie schnell, aber oberflächlich arbeitet; es fehlt der Kontrast. Mag der resultierende Weichzeichnereffekt mitunter erwünscht sein, so stört er doch, wenn ein ausdrucksstarkes Bild kontrastreich sein soll.

Dem läßt sich beikommen, indem man die Linse durch eine 35 mm-Kameralinse ersetzt; hierbei ist ein C-Adapter zum Befestigen notwendig. Zoom-Objektive sind für Copy-Stands weniger gut geeignet, da sie langsam arbeiten und teuer sind. Statt dessen benutzt man eine Linse mit fester Brennweite; die ideale Brennweite liegt bei gewöhnlichen Linsen zwischen 35-50 mm und bei 16 mm-Film- und Videolinsen zwischen 12 und 25 mm.

Die Kamera ist der Star in der Digitizingshow. Darüber darf jedoch ein kleines Utensil wie die Linsenkappe nicht unterschätzt werden. Es schützt nämlich nicht nur vor Staub, sondern hat ferner die Funktion, eine unbeabsichtigte Belichtung der lichtempfindlichen Leuchtstoffteilchen an der Aufnahmeröhre zu verhindern. Werden die Leuchtstoffteilchen auch nur für kurze Zeit intensivem Licht ausgesetzt, prägen oder brennen sich die Konturen der Lichtquelle ein. Beim Digitalisieren vermeide man also starke Lichtreflektionen, die zu Überwärmung führen könnten. Wichtig ist, daß man eine Kamera ohne Linsenkappe niemals direkt auf eine Lichtquelle richtet.

Wie wäre es mit einer Kamera, die praktisch unverwüstlich ist, scharfe Kontraste realisiert, nicht einbrennt, keine RGB-Filter benötigt, und leichter ist als ihre eigene Linse? Zu günstigen Preisen bieten Sony, Canon, Sanyo und Panasonic, um Beispiele zu nennen, Schwarzweiß- und Farbkameras mit CCD (Charged Coupled Device) an, die sich höchst solide verhalten (einschließlich der Aufnahmeröhre).

Mit CCD läßt sich ein konturenscharfes Bild erzeugen, das zum Digitalisieren hervorragend geeignet ist. Wer seine Linsenkappe mit chronischer Regelmäßigkeit verliert, weiß es zu schätzen, daß den Chip eine hohe Beständigkeit gegen Verbrennung und Nachhinken auszeichnet. Wenn das Beste vom Besten gerade gut genug ist, wählt man eine RGB-Videokamera. Da man die roten, grünen und blauen Signale direkt aus der Kamera und nicht über einen Farbfilter erhält, läßt sich die Qualität des Digitalisierergebnisses deutlich verbessern. Die meisten Kameras, die in Studios und in der Industrie zum Einsatz kommen (einschließlich CCDs), erzeugen RS-170 verschlüsselte Videosignale, die sich über Schalter justieren lassen. Einige Modelle sind zum Digitalisieren besonders geeignet, da sie über separate Ausgänge für die Farbsignale verfügen. Egal, wie man an die RGB-Information gelangt, werden die Bilder in jedem Fall heller, klarer und schärfer als die, die sich bei Verwendung von Filtern ergeben.

Wer erst einmal mit den Grundlagen des Digitalisierens vertraut ist und sich die Erfahrungswerte von Profis zueigen gemacht hat, dem werden die erforderlichen Handgriffe schnell zur Routine. Mit entsprechender Ausstattung und nach gewissenhaften Tests sind der Kunst der Reproduktion und Retusche kaum Grenzen gesetzt.

(Gene Brawn/deutsch von S. Bröker)

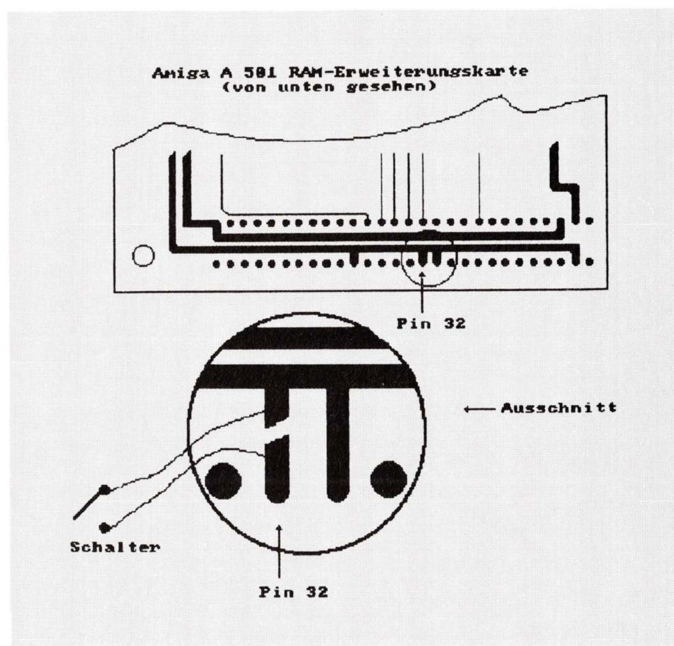
Wenn man den Speicher nicht gebrauchen kann

Umsteigern vom C64 werden Amigas 512 KB als unerschöpfliche Speichermenge vorkommen. Recht schnell stellt sich jedoch heraus, daß die besonderen Eigenschaften dieses Rechners ihren Preis haben — nicht in barer Münze, sondern in Form von blankem Speicher, der bei einigen Aufgaben wieder zum Problem wird.

Spätestens bei einigen Spezialfunktionen von DPaint II zeigen sich die engen Speichergrenzen eines 512-KB-Amiga. Stencil, Perspective, Background und Spare funktionieren erst ab 1 MB Arbeitsspeicher so richtig.
Zu den Speicherfressern zählen auch alle Animations-, DTP-, CAD- und Datenbankprogramme. Will man mit solchen Programmen arbeiten, dann heißt die Devise: Eine Speichererweiterung muß her! Das hat man auch bei Commodore erkannt und bietet für den Amiga 500 die RAM-Erweiterungskarte A501 an. Deren Installation ist einfach und problemlos. Doch danach beginnen die Probleme, wenn man unsauber programmierte Software aus der Steinzeit des Amiga verwenden will. Solche Software verträgt nämlich nur Arbeitsspeicher bis 512 KByte.
Für ein Programm muß vom Programmierer ein entsprechender Speicherbereich reserviert werden. Dazu wird meist die AllocMem-Routine aus der Exec-Library verwendet. Dieser muß mitgeteilt werden, welche Art von Speicher reserviert werden soll (siehe auch Memory-Map).
Mit dem Parameter MEMF__CHIP wird ein Speicherblock belegt, welcher von den Spezialchips erreicht werden kann. Derzeit sind das die untersten 512 KByte des Amiga.
Die Spezialchips (Agnus, Denise und Paula) können nur 512 KByte adressieren. Daher muß bei der Programmierung darauf geachtet werden, daß alle Daten, auf die diese Chips zugreifen, in den ersten 512 KByte untergebracht werden. Daran hielten sich, als der Amiga noch mit 256 KByte ausgestattet war, nicht alle Programmierer. So sind heute noch Programme (meist Spiele) in Umlauf, die mit einer Speichererweiterung nicht funktionieren. Um diesem Übel abzuhelfen, wurden von Zubehörfirmen abschaltbare Speichererweiterungen in den Handel gebracht. Wie hilft man sich aber bei der Commodore-A501, die ja nicht abschaltbar ist? Hier gibt es sowohl den „weichen“ wie auch den „harten“ Weg.

Zwang durch Doppelklick

Zuerst der weiche Weg: Auf der Workbench gibt es in der Schublade System das Programm NoFastMem. Ein Doppelklick auf dessen Icon zwingt den Amiga, nur Chip-RAM zu verwenden. Dieses Icon arbeitet wie ein Umschalter. Wenn wieder der volle Speicher genutzt werden soll, genügt ein erneuter Mausklick auf das Icon. Von der Shell oder dem CLI kann der volle Speicher wieder aktiviert werden, indem an das Programm NoFastMem eine Unterbrechung gesendet wird. Das kann entweder mit dem BREAK-Befehl oder durch die



Memory Map	
$000000	512 KByte Chip-RAM
$080000	Spiegelung des Chip-RAMs
$020000	8 MByte Fast-RAM
$A00000	CIA B und CIA A
$C00000	CPU-RAM 512 KByte Erweiterungen Amiga 500 und Amiga 2000
$C80000	leer
$DC0000	Echtzeituhr A500/2000
$DF0000	Register der Custom-Chips
$E00000	leer
$E80000	Expansionslots
$F00000	ROM-Module
$F80000	Spiegelung des Kickstart/ROMs
$FC0000	Kickstart/ROMs

Tastenfolge „Control C" erfolgen. Diese wirkt allerdings nur dann, wenn NoFastMem nicht im CLI durch RUN gestartet wurde.
Spiele werden aber meist direkt gebootet, das heißt ohne Workbench. Man hat in diesem Fall keine Möglichkeit, NoFastMem zu aktivieren.
Mit diesem Thema haben sich etliche Programmierer beschäftigt. Bei Durchsicht der PD-Kataloge finden sich mehrere einschlägige Programme.

Vorsicht nach der Behandlung

Das raffinierteste davon dürfte wohl FIXHUNK auf der Fish 36 sein. Dieses Programm ändert ein exekutierbares File so, daß DATA- und BSS-Hunks nur in das Chip-RAM geladen werden. CODE-Hunks werden nach wie vor in das Fast-RAM (falls vorhanden) geladen. Nach der Behandlung mit FIXHUNK laufen zum Beispiel die bekannte Diaschau „Dpslide" und das Demoprogramm „Robo-City" auch mit 1 MByte einwandfrei. Doch Vorsicht: FIXHUNK überschreibt das alte File. Also nur Programmkopien damit behandeln. Schonender geht das resetfeste NofastMem aus der AmigaWelt 4/88 (Seite 38) mit der Software um. Originale bleiben im Urzustand, dennoch kann gebootet werden, da der Speicher bei einem Reset abgeschaltet bleibt.
Weitere PD-Programme, die zur ersten Generation der No-Fast-Programme zählen und wie NoFastMem anzuwenden sind: Forget (Panorama 13c), Hide (Panorama 15c).
Zur zweiten Generation gehört StealMemBoot, und dieses befindet sich auf der Fish 150. Dieses Programm ist insofern bemerkenswert, als es den Bootblock modifiziert und damit StealMemBoot bereits beim Booten wirksam wird. Doch Vorsicht: Manche kommerziellen Programme verwenden keinen Standard-Bootblock. Hier darf StealMemBoot nicht angewendet werden!
Der „weiche" Weg löst fast immer auftretende Kompatibilitätsprobleme. Wer die Speichererweiterung trotzdem abschaltbar machen will, muß den „harten" Weg gehen, das heißt an der Hardware herumbasteln. Dagegen spricht hauptsächlich: Es geht jeglicher Garantieanspruch verloren. Ferner ist das nur geübten Lötkolbenartisten zu empfehlen, die zusätzlich Erfahrung im Umgang mit Platinen haben. Der Co-Autor ist gelernter Elektroniker und weiß, wovon er spricht.
Für alle, die es trotz dieser Warnungen versuchen wollen, folgt eine kurze, nicht sehr ausführliche Beschreibung. Kurz deswegen, um Anfänger abzuschrekken:
Um die RAM-Erweiterungskarte A501 abschaltbar zu machen, ist lediglich der Einbau eines Schalters erforderlich. Die Abbildung zeigt die Platine von der Unterseite. Hier ist die Leiterbahn, die von GROUND zum Pin 32 (EXRAM-Leitung) führt aufzutrennen und mit einem Schalter zu überbrücken. Im Gehäuse des Amiga ist auf der Rückseite genügend Platz für den Einbau des Schalters.
Der Schalter darf nur bei ausgeschaltetem Amiga betätigt werden. Nur wenn er geschlossen ist, erkennt der Amiga die Speichererweiterung. Ist der Schalter offen, dann ist die Speichererweiterung abgeschaltet. Die Hardware-Uhr bleibt davon unbeeinflußt und läuft unabhängig von der Schalterstellung weiter.
(rw)

Mit Super-VHS mehr Bildqualität für Hobby-Cutter

Starken Aufzeichnungsverlusten bei Videorecordern wollen Gerätehersteller mit einem neuen Industriestandard vermeiden. Das hat Folgen auch für die Computervideographen.

Die Produzenten von Videogeräten haben sich in den letzten Jahren mehr oder weniger auf den Standard VHS (Video Home System) geeinigt, das heißt, Grundig und Philips haben die Produktion ihrer Video-2000-Geräte eingestellt, wie es damals hieß aus Kostengründen, und im Laufe des letzten Jahres hat sich auch Sony, der Vorreiter und spätere Einzelkämpfer in bezug auf das Videosystem Betamax, entschlossen, neben der neuen Norm „Video 8" auch VHS-Geräte anzubieten.

In den Jahren zuvor boten die großen Videomarken der japanischen Konzerne sowie auch die europäischen Hersteller zu jeder Messe eine Neuigkeit, an dieser Stelle seien nur kurz die Begriffe Long Play, Stereo und High Quality Bildverbesserung (HQ) erwähnt, diese Neuerungen brachten aber unter dem Strich nie große Schritte in Richtung einer deutlich gesteigerten Bildqualität. Vielmehr waren es immer nur technische Kniffe, die dem Kunden ein besseres Bild vorgaukelten, physikalisch hatte sich an der Aufzeichnungsmethode nichts geändert. Die einzige wahre Änderung des VHS-Standards war im Zuge der Einführung der HiFi-Videorekorder erfolgt. HiFi-Videorekorder zeichnen das Tonsignal nicht mehr mit dem feststehenden Tonkopf auf, sondern mit dem rotierenden Videokopf. Aus Kompatibilitätsgründen zeichnen HiFi-Videorekorder allerdings auch noch die Toninformation mit dem starren Tonkopf auf, ansonsten könnten die Besitzer eines Videorekorders ohne HiFi-Option die in HiFi-Norm aufgezeichneten Bänder nur ohne Ton wiedergeben. Die Norm zur Bildaufzeichnung blieb unverändert.

VHS verschluckt 80 Zeilen

Bis heute waren Heimvideoaufzeichnungen immer schlechter als die von den Fernsehsendern gesendeten Bilder. Dies läßt sich durch die Aufzeichnungsgenauigkeit erklären, mit der normale Heimvideorekorder arbeiten. Ein Fernsehbild besteht aus 625 Zeilen, diese 625 Zeilen werden in zwei sogenannten Halbbildern wiedergegeben. Bei dem ersten Durchgang (dem ersten Halbbild) zur Darstellung eines Bildes werden die ungeraden Zeilen übertragen, im zweiten Durchgang die geraden Zeilen (das zweite Halbbild). Rechnerisch ergeben sich so pro Halbbild 312,5 Zeilen, de facto sind dies allerdings zirka 330 Zeilen, denn zu den 312,5 Zeilen für den eigentlichen Bildaufbau gesellen sich noch zirka 15 Zeilen für die sogenannte Austastlücke, die zur Synchronisation des Bildes durch den Fernseher genutzt wird. Heute werden in der Austastlücke auch Zusatzinformationen übertragen, so zum Beispiel der Videotextdienst der öffentlich rechtlichen Anbieter.

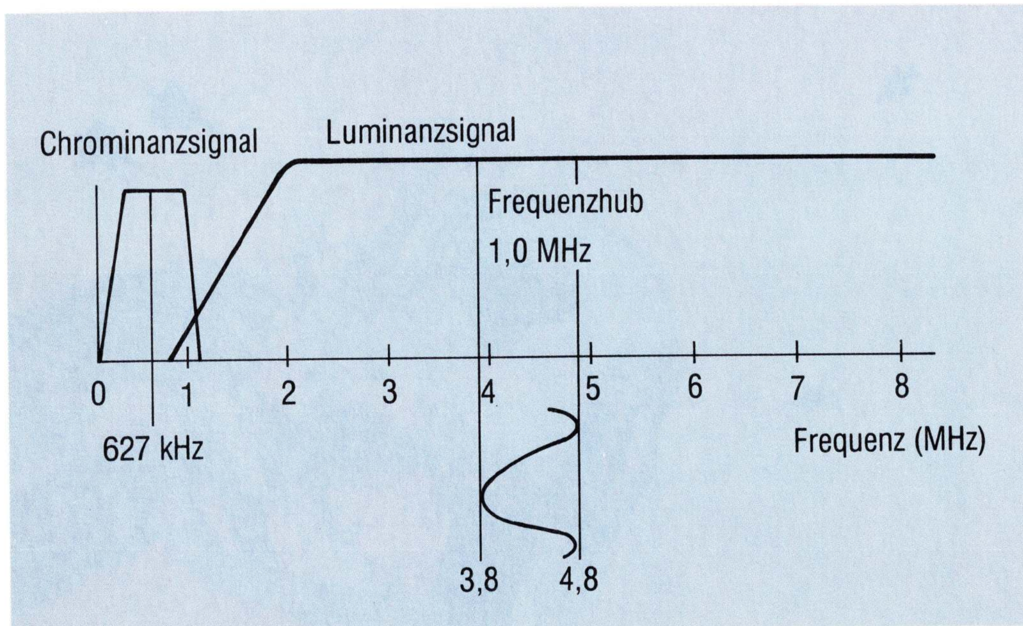
Von den zirka 330 Zeilen eines Fernsehhalbbildes zeichnet ein herkömmlicher VHS-Rekorder nur zirka 250 Zeilen auf, so erklärt sich also der Qualitätsverlust bei Videoaufzeichnungen.

Super-VHS zeichnet nun zirka 400 Zeilen eines Halbbildes auf, das heißt, es könnte mehr an Information aufgezeichnet werden, als überhaupt gesendet wird. Hieraus folgt, daß Aufzeichnungen mit einem Super-VHS-Rekorder kaum Qualitätsverlust zeigt. Gleichzeitig mit der Erhöhung der horizontal aufgezeichneten Signale wurde eine Trennung des Luminanz- und Chromasignals, das heißt einer Trennung der Helligkeits- und Farbsignale, durchgeführt. Hierdurch treten Interferenzstörungen dieser beiden Signale nicht mehr auf, eine verbesserte Farbwiedergabe ist die Folge. Folgerichtig ist natürlich, daß nun auch Fernseher mit dieser Trennung der Signale arbeiten, denn ansonsten würden auf dem Transport der Signale vom Videorekorder zum Fernseher Qualitätseinbußen auftreten. Der Käufer sollte daher beim Kauf eines neuen Fernsehgerätes heute schon darauf achten, daß der Fernseher über einen Super-VHS-Eingang verfügt.

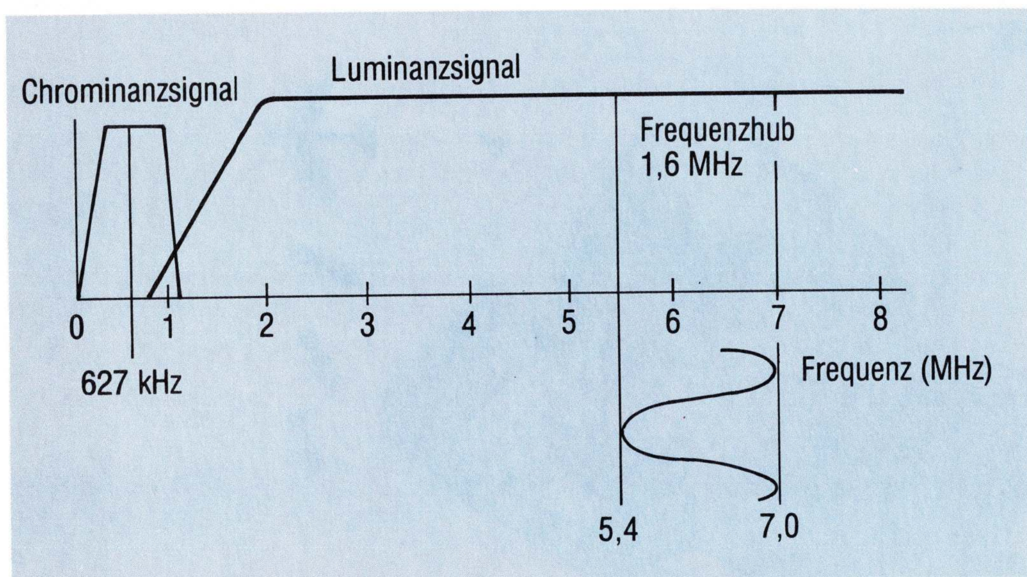
Auch das Überspielen beziehungsweise das Nachbearbeiten einer Videoaufzeichnung bietet nun eine verbesserte Qualität, vorausgesetzt beide Videorekorder verfügen über die entsprechenden Super-VHS-Ein- und -Ausgänge. Zum einen gehen bei der erneuten Aufzeichnung weniger Bildinformationen verloren, zum anderen sind die Übertragungsverluste deutlich minimiert, denn Farb- und Helligkeitssignal werden nicht mehr vom Sender gemischt und anschließend beim Empfänger wieder getrennt, sondern direkt getrennt übertragen.

Secam-Norm ist inklusive

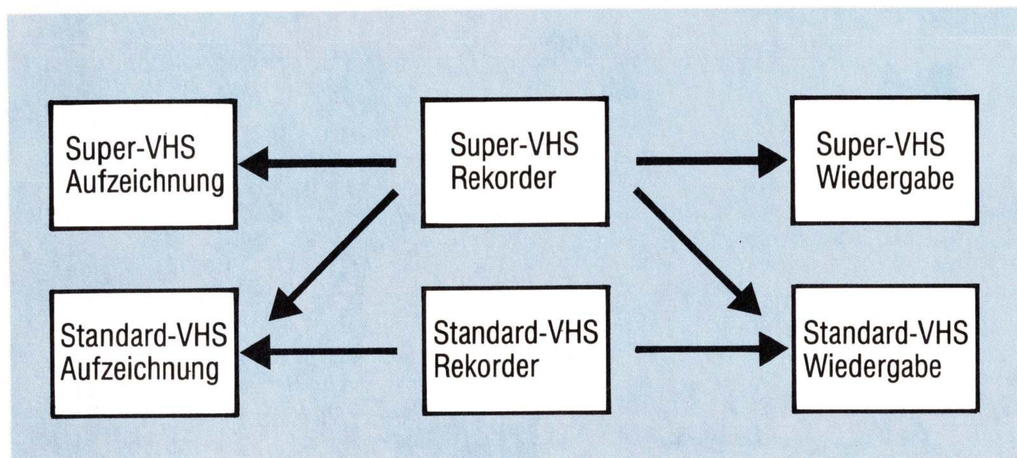
Darüber hinaus wurde das Trägerfrequenzband des Farbsignals von 3,8 MHz bis 4,8 MHz auf 5,4 MHz bis 7,0 MHz angehoben, hierdurch wird neben der Erhöhung der vertikalen (Zeilen-)Auflösung eine wesentliche Erhöhung der horizontalen (Spalten-)Auflösung erreicht, die auch eine wesentliche Bildverbesserung mit sich bringt (siehe auch Grafik). Hieraus resultiert, daß Super-



Standard VHS-Frequenzen-Spektrum



Super-VHS-Frequenz-Spektrum



Kompatibilität der Video-Systeme

VHS-Aufzeichnungen nur mit speziell für diese Aufzeichnungsdichte entwickeltem Bandmaterial möglich sind. Herkömmliches Bandmaterial ist hierfür nicht geeignet.
Zusätzlicher Luxus des Super-VHS-Standards ist, daß Aufzeichnungen in PAL- oder SECAM-Norm keine Unterschiede mehr aufweisen, das heißt, zum Beispiel in Frankreich aufgezeichnete Videos können in der Bundesrepublik auf einem PAL-Videorekorder wiedergegeben werden und umgekehrt.

Auf der abgebildeten Grafik können Sie sehen, daß aber nun niemand, der sich einen Super-VHS-Rekorder zulegt, von der Welt der normalen VHS-Software abgeschnitten ist, denn Super-VHS-Rekorder können auch in Standard-VHS aufgezeichnete Sendungen wiedergeben, allerdings ist der umgekehrte Weg nicht möglich. Super-VHS-Aufzeichnungen sind mit einem Standard-VHS-Rekorder nicht wiederzugeben.
Nach so viel Positivem, soll nun aber auch die Kehrseite der Medaille nicht verschwiegen werden. Super-VHS-Videorekorder sind annähernd doppelt so teuer wie vergleichbare Standard-VHS-Rekorder.

Der Amiga-Besitzer wird die vollen Vorteile von Super-VHS erst dann ausnutzen können, wenn die ersten Genlocks mit einem Super-VHS-Ein- und -Ausgang auf den Markt kommen, diese Eigenschaften werden allerdings vornehmlich bei teuereren Geräten zu finden sein. Aber auch bei normalen Aufzeichnungen vom FBAS-Ausgang des Amigas dürfte eine deutliche Qualitätssteigerung in der anschließenden Wiedergabe zu verzeichnen sein.

(Stefan Jägers/Volkmar Eich)



Amiga und die Viren

„Cave canem!", sagten die alten Lateiner; heute heißt es: „Hüte Dich vor dem Virus." Wo immer die Viren auch herkommen, betroffen sind alle. Profis wissen, wie man Viren erkennt und fachmännisch eliminiert. Für Einsteiger aber ist das ein Problem.

Über die Herkunft der Viren gibt es die wildesten Gerüchte. Tatsache ist, daß Viren auch nur ganz normale Programme sind und sie daher zunächst einmal gestartet werden müssen. Da bietet sich der Boot-Vorgang, der bei jedem Kaltstart oder Reset ausgeführt wird, geradezu an. Hierbei werden die Bootblöcke, das sind die ersten beiden Blöcke auf der Diskette, eingelesen und ab dem dritten Langwort als Programm abgearbeitet. Normalerweise bekommt der Benutzer deren Inhalt nie zu Gesicht. Aus diesem Grund nutzen die sogenannten Bootblock-Viren diese Möglichkeit, in das System zu gelangen, wo sie eine Kopie ihres Programmcodes ablegen. Damit dieser Code automatisch aktiviert wird, initialisieren sie einen Zeiger auf sich selbst. Beim Amiga bieten sich dafür die „ColdCapture"- und „CoolCapture"-Vektoren an, weil diese bei jedem Reset angesprungen werden.

Bevor sich solch ein Virus vermehren kann, muß es erst überleben; daher meldet es dem Betriebssystem den von ihm belegten Speicherbereich als belegt. Zur Vermehrung nutzt das Virus aus, daß bei jedem Bootvorgang eine Diskette im Laufwerk liegt. Über das Trackdisk-Device kopiert es sich auf diese Diskette. Dadurch ist auch diese verseucht.

Der Programmteil des Virus kann die unterschiedlichsten Funktionen ausführen. Zwei Wirkungsarten lassen sich unterscheiden: Zum einen die Manipulationen von Daten (Löschen oder Verändern von Programmen), zum andern die Behinderung der Arbeit.

Viren können so programmiert sein, daß der Anwender auftretende Fehler anfangs für Hardware- oder Bedienungsfehler hält. Einsteigern, die mit dem Amiga noch nicht sehr vertraut sind, gibt das unlösbare Rätsel auf.

Urvater aller Amiga-Viren ist das SCA-Virus. Dieses kopiert sich selbständig auf den Bootblock von Disketten und gibt bei jedem 16. Reset mit der Tastenkombination CTRL/Amiga/Amiga eine Meldung aus, die so beginnt: „Something wonderful has happened... Your Amiga is alive..." Ferner wird der verblüffte Anwender darauf hingewiesen, daß einige Disketten mit Viren infiziert sein könnten.

Das SCA-Virus ist nicht so harmlos, wie es zunächst aussieht, denn es kopiert sich in die Bootblöcke der eingelegten Diskette. Viele kommerzielle Programme (meist Spiele) haben keinen Standard-Bootblock, sondern legen den Kopierschutz in die Bootsektoren. Bei einer verseuchten Diskette kann dieser nicht mehr abgefragt werden und das Programm wird unbrauchbar.

Es gibt eine Reihe von Abkömmlingen des SCA-Virus, die noch unangenehmer sind. So werden statt der Meldung unerwartete Resets ausgelöst. Eine besonders gefährliche Abart formatiert den Cylinder 40 der Diskette. Das ist deshalb besonders schlimm, weil dadurch das Directory der Diskette gelöscht wird.

Link-Viren hängen sich an vorhandene Programme. Sie können sich im Disk-Validator oder dem RAM-Handler verstecken. Auch CLI-Befehle werden von ihnen als Versteck benutzt. Wenn von Zeit zu Zeit verräterische Meldungen auftauchen und das Virus in den Bootblökken nicht zu finden ist, dann ist wahrscheinlich ein Link-Virus die Ursache.

Kaum zu killen, weil nur schwer zu finden, ist das UCAIDS-Virus, weil es sich bei jedem Schreiben verändert.

Ganz raffiniert tarnt sich das North-Star-Virus. So paradox es klingt, es hat einen Viren-Detektor eingebaut und erkennt das SCA-Virus und das Byte-Bandit-Virus.

Auch Disketten mit „Bad Sectors" sind verdächtig, weil es möglich ist, Sektoren, in denen sich das Virus befindet, als „Bad Sector" zu kennzeichnen und es damit weitgehend vor Entdeckung zu schützen.

Präventivmaßnahmen gegen Byte-Bandits

Am erfolgreichsten ist derzeit neben dem SCA-Virus das Byte-Bandit-Virus. Es äußert sich folgendermaßen:

Der Bildschirm ist plötzlich komplett leer. Alles, was bleibt, ist ein blauer oder dunkler Schirm. Scheinbar geht jetzt nichts mehr — außer einem Reset.

Temporäre Abhilfe bringt das gleichzeitige Drücken folgender Tasten:

Alt links
Amiga links
Leertaste
Amiga rechts
Alt rechts

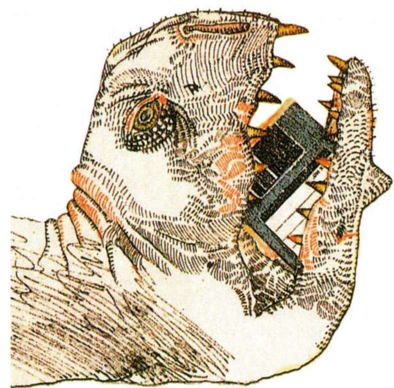
Falls gerade wichtige Daten im Speicher stehen, können diese dann rasch auf eine Diskette gerettet werden.

Trotz aller Spekulationen über die Fähigkeiten von Viren, bleibt die Tatsache unumstößlich, daß jeder Virus durch Abschalten des Computers garantiert abgetötet wird. Der Computer muß nur lange genug abgeschaltet werden, damit sich alle Kondensatoren der Stromversorgung restlos entladen können. Solange die RAM-Chips noch mit Strom versorgt werden, lebt das Virus. Messungen haben gezeigt, daß das einige Minuten dauern kann. In der Regel bieten jedoch zwei Minuten bereits mehrfache Sicherheit. Verseuchte Disketten werden natürlich dadurch nicht kuriert. Auf Disketten, Festplatten und in batteriegepufferten RAMs können Viren überleben.

Viren werden vorwiegend über (Raub)-Kopien verbreitet. Auch Software in Mailboxen kann verseucht sein. PD-Software wurde bisher nur selten infiziert. Wenn man also Programme tauscht oder kopiert, ist Vorsicht angebracht.

Steht ein Virus-Checker oder Virus-Killer zur Verfügung, so sollte jede unbekannte Diskette damit behandelt werden. Leider gibt es schon so viele Mutationen von Viren, daß auch damit keine absolute Sicherheit mehr gegeben ist. Man sollte sich daher angewöhnen, nach jeder fremden Diskette den Amiga komplett abzuschalten und neu zu starten. Ein Reset genügt nicht, weil Viren resetfest sind.

Ein guter Schutz ist es, ausschließlich schreibgeschützte Disketten in das Laufwerk einzulegen und den Schreibschutz nur dann zu desaktivieren, wenn es unbedingt sein muß. Eine schreibgeschützte Diskette ▶



kann nicht infiziert werden. Wird eine verseuchte Diskette in das Laufwerk eingelegt, kann sie den Speicher des Amiga verseuchen, und jede weitere (nicht schreibgeschützte) Diskette wird von dort aus infiziert. Weiteren Schutz bieten Sicherheitskopien. Im Ernstfall kann so der Datenbestand vom Original wieder erneuert werden. Originale dürfen daher niemals als Arbeitsdisketten benutzt werden.

Im Fachhandel werden Viren-Checker und Viren-Killer angeboten. Einige Softwarefirmen wittern das große Geschäft. Solche Programme sind in Sachen Vorbeugung recht hilfreich. Sie können aber nur bekannte Viren entdecken. Ein universelles Schutzprogramm wird es folglich nicht geben.

Günstig fährt man mit PD-Software. „VCheck" heißt der Virus-Detektor, der von Bill Koester (Commodore Amiga Technical Support) geschrieben wurde. Die Version 1.2 befindet sich auf der Fish-Disk #126. Davon gibt es inzwischen Updates. „VirusX" nennt sich ein Programm, welches Disketten sofort auf Viren überprüft. Dieser Virus-Checker läuft als Hintergrundprogramm und meldet sich nur, wenn ein Virus entdeckt wird. „VirusX" ist auf der Fish-Disk #137.

Auch deutsche Programmierer schlafen nicht. So wurde in der Amiga-Welt (Heft 2/88) ein „ViCheck" publiziert. Dieses ist auf der Leserdiskette 2/88 enthalten. Das Programm kann sowohl von der Workbench als auch vom CLI gestartet werden und läuft im Hintergrund. Noch besser, man bindet es mit dem bekannten PD-Programm „RunBack" in die Startup-Sequence ein. „ViCheck" meldet veränderte Bootblöcke und verbogene Vektoren. „ViCheck" entdeckt daher nahezu alle bisher bekannten Bootblock-Viren und resetfesten Speicherteile — leider auch RAD: und TurboPrint.

Glückliche Besitzer einer Workbench 1.3 haben es leichter, denn der verbesserte INSTALL-Befehl besitzt eine Check-Option.

Therapieversuche ohne Erfolgsgarantie

Falls der Virus-Checker Alarm schlägt oder ein Verdacht auf Virenbefall aufkeimt, sollte man sofort die Arbeit unterbrechen und den Computer abschalten. Allerdings können sich Viren-Checker irren. So halten manche Detektoren und Virus-Killer einwandfreie, aber Non-Standard-Bootblöcke für ein Virus. Arbeitet der Virus-Checker mit der Prüfsumme des Bootblocks, so kann beispielsweise das LSD-Virus nicht erkannt werden, weil sich ja mit der Änderung eines Bytes auch die Prüfsumme ändert.

Klarheit schafft hier ein Disketten-Monitor. Im PD-Angebot gibt es auf der Fish-Disk #71 den Disketten-Editor „DiskX", der für die Viren-Suche auch von Einsteigern problemlos bedient werden kann. Für diese folgt eine Bedienungsanleitung:

Von der Fish-Disk # kopiert man den DiskX am besten auf eine garantiert saubere Arbeitskopie der Workbench. Nachher ist deren Schreibschutz wieder zu aktivieren, und mit dem bekannten Doppelklick wird DiskX gestartet.

DiskX bietet einige Menüs, von denen für die Virensuche nur die Menüs „Disk" und „Show As" gebraucht werden. Mit dem Disk-Menü wird das Laufwerk eingestellt, in welchem die zu untersuchende Diskette eingelegt ist, und mit „Show As" stellt man am besten „Ascii Text ein.

Die Bootsektoren befinden sich, wie schon erwähnt, auf Block 0 und 1. Statt Block ist auch die Bezeichnung „Key" gebräuchlich und wird im DiskX verwendet. Nach dem Start zeigt dieses Feld auf Key 880. Hier muß 0 eingesetzt werden.

Dazu stellt man den Mauszeiger auf die erste Ziffer und klickt diese an. Dann wird die Del-Taste zweimal betätigt, und es steht nurmehr die gewünschte Null im Feld. Mit der Return-Taste wird die Änderung eingegeben. Als Ergebnis wird der Inhalt des Blocks in ASCII-Darstellung ausgegeben. Das Byte-Bandit-Virus erkennt man am Eintrag: „Virus by Byte Bandit in 9.87 Number of copys. . .". Das SCA-Virus verrät sich erst in Block 1. Dorthin gelangt man, indem der Aufwärts-Pfeil neben dem Keygadget angeklickt wird. Beginnt der Texteintrag mit „Something wonderful has happened. . .", dann hat sich das Original-SCA-Virus auf dieser Diskette eingenistet.

Nur wenn wirre Texte erscheinen, kann angenommen werden, daß die Bootsektoren sauber sind. Alle deutsch oder englisch lesbaren Intros machen die Diskette verdächtig.

Handelt es sich um das SCA-Virus oder einen seiner Abkömmlinge, so ist diesen mit folgender Therapie beizukommen:

1. Um ganz sicher zu gehen, den Amiga für zwei Minuten abschalten (Theoretisch würden auch 30 Sekunden genügen).
2. Den Amiga mit einer „sauberen" Workbench wieder hochfahren.
3. Ein CLI-Fenster öffnen und den INSTALL-Befehl in die RAM-Disk kopieren.
4. Die verseuchte Diskette in ein Laufwerk einlegen und den infizierten Bootblock mit ram:install dfX: überschreiben. Anstelle von „X" ist natürlich die Nummer des Laufwerks einzusetzen. Nur durch das Einlegen in das Laufwerk passiert nichts, denn das Virus wird nur durch den Bootvorgang in df0: aktiviert.

Kommerzielle Software mit modifizierten Bootblöcken kann durch diese Kur allerdings nicht gerettet werden. Auch auf das Byte-Bandit-Virus kann diese Kur nicht angewendet werden, weil sich dieses bei jedem Reset in den Bootblock zurückschreibt. Hier wirkt nur ein für dieses Virus geeigneter Virus-Killer oder ein Disketten-Monitor. Dazu ein Rezept für Leser, die mit einem Diskettenmonitor umzugehen wissen:

1. Um ganz sicher zu gehen, den Amiga für zwei Minuten abschalten.
2. Den Amiga mit einer „sauberen" Workbench, auf die der Disketten-Monitor kopiert wurde, wieder hochfahren.
3. Die infizierte Diskette in ein Laufwerk legen und Block 0 einlesen.
4. Den Texteintrag des Byte-Bandit-Virus im Hex-Modus mit Nullen oder im ASCII-Modus mit der Leertaste überschreiben und anschließend den Block wieder auf die Diskette zurückschreiben.
5. Die infizierte Diskette mit INSTALL behandeln, die Punkte 1 und 2 wiederholen und mit dem Virus-Checker und/oder Disketten-Monitor überprüfen, ob die Bootblöcke sauber sind.

Die beschriebenen Anti-Viren-Kuren wirken nur auf Bootblock-Viren. Wenn zukünftig Viren entwickelt werden, die nicht mehr auf die Bootblöcke angewiesen sind, dann beginnt ein neues Räuber- und Gendarm-Spiel, denn die Entwicklung der Viren ist noch lange nicht abgeschlossen.

Zum Abschluß noch ein paar Hinweise auf Link-Viren. Man wird nicht lange warten müssen, bis diese in den Amigas häufiger auftauchen. Großanlagen sind heute schon damit verseucht. Besteht der Verdacht auf einen Link-Virus, so sind alle in Frage kommenden Programme auf ihre Länge zu überprüfen. Besonders verdächtig ist der DIR-Befehl. Ein Hinweis für Einsteiger: Der LIST-Befehl des Amiga-DOS liefert die Programmlängen.

(iw/rw)

LESERSERVICE

Bestellen statt Abtippen! Für alle Leser, die sich mühsames Abtippen, Falscheingaben, Fehlerquellen, Testläufe und den ganzen Zeitaufwand sparen wollen: Hier sind alle Programme der AMIGA-WELT-Ausgaben auf Disketten. Natürlich auch die Programme aus diesem Heft! **Gleich bestellen!**

10 DM pro Diskette + 5 DM Versandkosten pro Bestellung. Keine Nachnahme!

AMIGA WELT 2/89

```
1> cd
AmigaWelt2|89

1> dir opt a
        check (dir)
        pics (dir)
        alle Listings dieser Ausgabe (dir)
1>
```

Commodore

Bestellnummer: 1/88a
- Compactor-Bild und Sound
- Animierte 3D-Stereo-Grafik
- Druckerroutine für Epson
- und 3 weitere Programme

Bestellnummer: 1/88b
- Listings zum Assemblerkurs
- Listings zum C-Kurs
- Hack-Adventure
- und 4 weitere Programme

Bestellnummer: 2/88
- Quickcopy-Kopierprogramm
- SlideShow für IFF-Bilder
- VICheck-Viruschecker
- und weitere 12 Programme

Bestellnummer: 3/88
- Color-Changer ändert Farbpaletten
- Grafik-einfaches Zeichenprogramm
- IconAssembler für Doppelicons
- und weitere 6 Programme

Bestellnummer: 4/88
- ShortSound-Tonkanalprogrammierung
- Nofastmem-Speicherabschaltung
- CopySmus-Sonix-Hilfsutility
- und weitere 3 Programme

Bestellnummer: 5/88
- Keymaps.asm-Tastaturorganisation
- SetPrefs-Preferenceseinstellungen
- Change-Konvertierungshilfe
- und weitere 2 Programme

Bestellnummer 1/89
- Funktionsplotter-Formen aus Formeln
- Soundeffekte für BASIC
- Patchen mit BASIC
- und weitere 6 Programme

Ausführliche Programmbeschreibungen und Anleitungen finden Sie in der jeweiligen Ausgabe der AMIGA-WELT. Die Bestellnummer der Diskette ist identisch mit der Ausgabe der AMIGA-WELT, in der das Programm vorgestellt ist.

Und so bestellen Sie:

Bestellschein deutlich ausfüllen und gewünschte Diskette(n) angeben. Pro Bestellung DM 5,- für Versandkosten dazurechnen. Gesamte Summe per Euroscheck mitschicken oder vorab auf das Konto Nr. 0730006344 bei der Sparkasse Berlin (West), BLZ 100 500 00 (Kontoinhaber: technicSupport GmbH), überweisen. Auslieferung erfolgt sofort nach Zahlungseingang!

Bestellungen an:



technicSupport
Marketing und Verlag GmbH
Bundesallee 36 - 37
1000 Berlin 31

Leser aus Österreich wenden sich an folgende Adresse:
m.a.r. – peter rauscher's COMPUTERSHOP, A-1100 Wien, Weldengasse 41, Tel. 0222/62 15 35

technicSupport

BESTELLSCHEIN

Ich bestelle bei technicSupport GmbH die Leserdisketten mit der Bestellnummer: ____________

Name: ____________

Straße: ____________

Ort: ____________

Unterschrift: ____________

Datum: ____________

2000 Mark für das



Zaubermeister verhext IFF-Grafiken

Selbst DPaint-Werke bedürfen manchmal der Nachbereitung. Das Electronic-Arts-Produkt stellt leider nicht alle notwendigen Utilities zur Verfügung. Speziell für IFF-Grafiken im Format 320×200 wurde Zaubermeister erschaffen. Das Programm enthält verschiedene Möglichkeiten der Bildmanipulation, wie auch eine Funktion zum Sortieren der Farben für das Programm IFF-Print (AmigaWelt 1/88).

Das Programm Zaubermeister wird aktiviert, indem man das Ladeprogramm „Meister" im AmigaBASIC startet. Die Files AmigaBASIC, Meister und Zaubermeister müssen dabei auf derselben Diskette sein. Zaubermeister benötigt die folgenden Libraries: exec.library, dos.library, graphics.library und intuition.library.

Wenn das Programm Meister geladen ist, gibt eine flimmernde Schrift einige Hinweise aus. Auf Tastendruck wird der Bildschirm schwarz, und der Amiga lädt Zaubermeister nach.
Die einzlnen Funktionen sind über Pull-Down-Menüs erreichbar. Es stehen folgende Funktionen zur Verfügung:

Bild laden: Ein IFF-Bild im Format 320×200 wird geladen. Der Filename wird extra abgefragt.

Bild speichern: Das Bild wird im IFF-Format gespeichert.

Farben ändern: Es wird ein Fenster geöffnet, und die 32 Farben werden oben als kleine Quadrate dargestellt. Links sind drei Schieberegler für die Grundfarben Rot, Grün und Blau. Anklicken der Felder unter den Buchstaben R, G, B bewirkt, daß der jeweilige Farbanteil vor Veränderungen geschützt ist. Möchte man eine Farbe ändern, so klickt man entweder das entsprechende Quadrat oben an, oder man klickt sie im Bild an. Die Regler stellen sich auf die neue Farbe ein, und man kann diese durch Verschieben der Regler ändern. Um zwei Farben zu mischen, wird das Feld „Misch" angeklickt. Nun muß die erste der beiden zu mischenden Farben angeklickt werden, danach die zweite. Die Mischfarbe ersetzt dann die zweite gewählte Farbe. Um diesen Menüpunkt zu verlassen, muß man das Feld „OK" an-



Rot-, Grün- oder Blauanteil können vor Farbmanipulationen fixiert werden.



Im Farbmenü wird das komplette Bild bearbeitet, unter „Clip" nur ein definierter Ausschnitt.

klicken. Gleiche Vorgehensweise gilt für die Funktionen Copy und Spread.

Drucken: Eine Hardcopy des Bildschirms wird erstellt.

Ende: Das Programm wird beendet.

Voreinstellung: Ein Fenster mit den 32 Farben wird geöffnet. Unter ihnen befindet sich am Anfang je ein „J". Klickt man die Farbe an, so wird das „J" zu einem „N". So kann man zulassen oder verhindern, daß die Farbe durch eine Farboperation verändert wird. In der Mitte des Fensters steht „Mosaik-X" und „Mosaik-Y". Durch Anklicken des „ + "- oder des „—"-Symbols erhöht oder erniedrigt man den Wert, der die X- oder Y-Größe des Mosaiks angibt. Beendet wird diese Option durch „OK".

Heller-Dunkler: Hier kann man die Helligkeitswerte aller zugelassenen Farben erhöhen beziehungsweise erniedrigen. Beendet wird die Option durch „OK".

Grautöne: Die zugelassenen Farben werden in entsprechende Grauwerte umgerechnet.

RGB-Töne: Die zugelassenen Farben werden wahlweise in entweder Rot-, Grün- oder Blauwerte geändert.

Verschwimmen: Die zugelassenen Farben werden einander „nähergebracht".

Kontrast: Die zugelassenen Farben werden „voneinander entfernt".

Antik: Die zugelassenen Farben werden in Antiktöne geändert.

Zufallsfarben: Die zugelassenen Farben werden nach Zufallsprinzip verändert.

Sortieren: Alle Farben werden entweder aufwärts oder abwärts sortiert. Dies dauert allerdings sehr lange und trägt auch nicht zur Lesbarkeit bei, ist aber für das Programm „IFF-Print" sehr nützlich.

RGB-Filter: Man kann aus den erlaubten Farben die RGB-Töne herausfiltern.

Komplementär: Die Komplementärfarben der erlaubten Farben werden errechnet.

Markieren: Der Ausschnitt wird markiert.

Löschen: Der Ausschnitt wird gelöscht.

Kopieren: Nachdem man eine neue linke obere Ecke angegeben hat, wird der Ausschnitt kopiert.

Verändern: Hier muß man sowohl eine neue linke obere Ecke sowie eine neue rechte untere Ecke angeben. Der Ausschnitt wird nun kopiert und gleichzeitig in der Größe verändert.

Spiegeln: Der Ausschnitt wird gespiegelt.

Kippen: Der Ausschnitt wird gekippt.

Invertieren: Der Ausschnitt wird invertiert.

Mosaik: Der Ausschnitt wird zu einem Mosaik gemacht.

Verschieben: Mit der Maus kann das Bild verschoben werden. Achtung, verläßt ein Teil des Bilds den Bildschirm, so ist es gelöscht! Mit einem Mausklick beendet man diese Option.

Die Funktionen Mosaik, Spiegeln, Kippen, rechte Seite spiegeln, linke Seite spiegeln und Invertieren im Menüpunkt Effekte haben die gleiche Funktion wie die Funktionen im Menüpunkt Clip, jedoch beziehen sie sich auf das gesamte Bild.

Wenn „Zaubermeister" mit AmigaBASIC eingegeben wird, muß mit „CLEAR 60000,20000" ausreichend Speicherplatz reserviert werden. Das Ladeprogramm „Meister" kann ohne Probleme abgetippt werden.

Zaubermeister verarbeitet in der vorliegenden Fassung nur NTSC-Bilder. Auf der Leserdiskette befindet sich zusätzlich eine Version für PAL-Bilder, die auch die Möglichkeit aufweist, zwei Farben auf ein Palette-Feld zu schreiben.

(A. Neumann/D. Rädel)

Programmname: Zaubermeister
Sprache : AmigaBASIC

Utilities: exec.bmap, intuition.bmap, dos.bmap, graphics.bmap und das AmigaBASIC müssen sich im root-Verzeichnis befinden



Auf Leserdiskette



Eingabe mit beliebigem Editor (ohne Zeilennummern)
Überprüfen mit checksum (neue Version)

```
CLEAR,60000&,20000                                 997
WINDOW CLOSE 1                                     480
SCREEN 1,320,200,5,1                               598
WINDOW 2,"ZAUBERMEISTER c 1988 by NEUDELSOFT",,16,1   365
PALETTE 0,0,0,0                                    980
PALETTE 1,1,1,1                                    715
PALETTE 2,1,1,1                                    087
COLOR 2,0                                          720
a$=INKEY$                                          026
FOR a=1 TO 8                                       843
 READ text$                                        600
 LOCATE 10,20-LEN(text$)/2:PRINT text$             982
 FOR b=0 TO 100 STEP 2                             054
  PALETTE 2,ABS(COS(b)),ABS(SIN(b)),(b/100)        589
  warte                                            632
  GOSUB schau                                      596
 NEXT b                                            485
 IF fo=1 THEN a=8                                  424
 FOR b=100 TO 0 STEP -2                            037
  PALETTE 2,ABS(COS(b)),ABS(SIN(b)),(b/100)        824
  warte                                            447
  GOSUB schau                                      263
 NEXT b                                            832
 IF fo=1 THEN a=8                                  427
 LOCATE 10,1:PRINT SPACE$(40)                      581
NEXT a                                             245
IF fo=1 THEN CALL boot                             494
LOCATE 20,1:PRINT "Bitte drücken Sie einen Taste"  644
LOCATE 1,1:PRINT "ZAUBERMEISTER"                   179
ax%=0:ay%=0                                        020
xd%=1:yd%=1                                        437
warte2:                                            068
IF ax%+xd%>208 THEN xd%=-xd%                       376
IF ax%+xd%<1 THEN xd%=-xd%                         545
IF ay%+yd%<1 THEN yd%=-yd%                         166
IF ay%+yd%>143 THEN yd%=-yd%                       724
SCROLL (ax%,ay%-1)-(ax%+(13*8),ay%+8),xd%,yd%      698
ax%=ax%+xd%:ay%=ay%+yd%                            113
```

```
GOSUB schau                                          481
IF fo=1 THEN CALL boot                               914
GOTO warte2                                          024
DATA "Neudelsoft proudly presents"                   026
DATA "Z A U B E R M E I S T E R"                     370
DATA "written by A.Neumann in 1988"                  150
DATA "mit Hilfe von Routinen von"                    512
DATA "Carolyn Scheppner CBM"                         296
DATA "Greetings to :"                                691
DATA "DANNY,Swertz,Hartmann,G.F.,Julia,Bella"        509
DATA "and the whole 9a"                              576
SUB warte STATIC                                     517
 FOR c=1 TO 50:NEXT c                                777
END SUB                                              578
SUB boot STATIC                                      292
 FOR a=0 TO 31:PALETTE a,0,0,0:NEXT a                744
 WINDOW CLOSE 2                                      620
 CHAIN "zaubermeister"                               369
END SUB                                              330
schau:                                               528
 IF INKEY$<>"" THEN fo=1                             371
 RETURN                                              907
                                                     907
                                                     907
                                                     907
```

Mit diesem kleinen Programm wird das Hauptprogramm nachgeladen.

```
REM ***************************************          000
REM *            Zaubermeister              *          000
REM *                von                    *          000
REM *          Andreas Neumann              *          000
REM *          7997 Immenstaadsec           *          000
REM ***************************************          000
REM                                                    000
secMain:                                               414
DIM bPlane&(5), cTabWork%(32), cTabSave%(32)           074
ccrtDir%   = 0                                         043
ccrtStart% = 0                                         259
ccrtEnd%   = 0                                         256
ccrtSecs&  = 0                                         753
ccrtMics&  = 0                                         028
DECLARE FUNCTION xOpen&  LIBRARY                       179
DECLARE FUNCTION xRead&  LIBRARY                       588
DECLARE FUNCTION xWrite& LIBRARY                       384
DECLARE FUNCTION AllocMem&() LIBRARY                   588
DECLARE FUNCTION AllocSignal%() LIBRARY                918
DECLARE FUNCTION FindTask&()     LIBRARY               004
DECLARE FUNCTION DoIO&()         LIBRARY               269
DECLARE FUNCTION OpenDevice&     LIBRARY               024
LIBRARY "dos.library"                                  810
LIBRARY "exec.library"                                 255
LIBRARY "graphics.library"                             652
LIBRARY "intuition.library"                            659
GOTO vorbereitungen                                    323
neuladen:                                              185
REM *** IFF - Laderoutine by Carolyn Scheppner  CBM
 ***                                                   185
ccrtDir%   = 0                                         002
ccrtStart% = 0                                         322
ccrtEnd%   = 0                                         911
ccrtSecs&  = 0                                         348
ccrtMics&  = 0                                         707
CLS                                                    001
loadError$ = ""                                        720
GOSUB LoadILBM                                         036
IF loadError$ <> "" THEN GOTO Mcleanup                 972
IF foundCCRT AND ccrtDir% THEN                         609
   FOR kk = 0 TO nColors% -1                           893
      cTabSave%(kk) = PEEKW(colorTab&+(kk*2))          026
      cTabWork%(kk) = cTabSave%(kk)                    277
   NEXT                                                956
   FOR kk = 0 TO 80                                    929
      IF ccrtDir% = 1 THEN                             665
         GOSUB Fcycle                                  763
      ELSE                                             020
         GOSUB Bcycle                                  634
      END IF                                           024
      CALL
       LoadRGB4&(sViewPort&,VARPTR(cTabWork%(0)),nC
       olors%)                                         455
      FOR del = 0 TO ccrtSecs& * 3000                  757
         FOR de2 = 0 TO ccrtMics& / 500                641
         NEXT                                          464
      NEXT                                             679
   NEXT                                                342
   CALL
    LoadRGB4&(sViewPort&,VARPTR(cTabSave%(0)),nColo
    rs%)                                               081
END IF                                                 631
Mcleanup:                                              346
FOR de = 1 TO 500:NEXT                                 433
Mcleanup2:                                             694
IF loadError$ <> "" THEN PRINT loadError$              787
IF saveError$ <> "" THEN PRINT saveError$              442
GOTO endeladen                                         555
Bcycle:                                                903
cTemp% = cTabWork%(ccrtEnd%)                           341
FOR jj = ccrtEnd%-1 TO ccrtStart% STEP -1              859
   cTabWork%(jj+1) = cTabWork%(jj)                     290
NEXT                                                   513
cTabWork%(ccrtStart%) = cTemp%                         026
RETURN                                                 034
Fcycle:                                                478
cTemp% = cTabWork%(ccrtStart%)                         443
FOR jj = ccrtStart%+1 TO ccrtEnd%                      393
   cTabWork%(jj-1) = cTabWork%(jj)                     834
NEXT                                                   569
cTabWork%(ccrtEnd%) = cTemp%                           023
RETURN                                                 015
LoadILBM:                                              137
f$ = ILBMNam$                                          068
fHandle& = 0                                           001
mybuf& = 0                                             139
foundBMHD = 0                                          519
foundCMAP = 0                                          849
foundCAMG = 0                                          778
foundCCRT = 0                                          359
foundBODY = 0                                          598
filename$ = f$ + CHR$(0)                               199
fHandle& = xOpen&(SADD(filename$),1005)                845
IF fHandle& = 0 THEN                                   732
   loadError$ = "ILBM-Datei nicht gefunden/lesbar."    663
   GOTO Lcleanup                                       168
END IF                                                 854
ClearPublic& = 65537&                                  323
mybufsize& = 360                                       833
mybuf& = AllocMem&(mybufsize&,ClearPublic&)            244
IF mybuf& = 0 THEN                                     068
   loadError$ = "Pufferspeicher nicht verfügbar."      897
   GOTO Lcleanup                                       490
END IF                                                 008
inbuf& = mybuf&                                        684
cbuf& = mybuf& + 120                                   414
ctab& = mybuf& + 240                                   357
rLen& = xRead&(fHandle&,inbuf&,12)                     089
tt$ = ""                                               706
FOR kk = 8 TO 11                                       405
   tt% = PEEK(inbuf& + kk)                             988
   tt$ = tt$ + CHR$(tt%)                               175
NEXT                                                   990
IF tt$ <> "ILBM" THEN                                  038
   loadError$ = "Keine Standard-ILBM-Grafikdatei."     161
   GOTO Lcleanup                                       650
END IF                                                 936
ChunkLoop:                                             269
 rLen& = xRead&(fHandle&,inbuf&,8)                     618
 icLen& = PEEKL(inbuf& + 4)                            839
 tt$ = ""                                              880
 FOR kk = 0 TO 3                                       020
    tt% = PEEK(inbuf& + kk)                            975
    tt$ = tt$ + CHR$(tt%)                              974
 NEXT                                                  365
IF tt$ = "BMHD" THEN  'BitMap-Header                   503
   foundBMHD = 1                                       476
   rLen& = xRead&(fHandle&,inbuf&,icLen&)              486
   iWidth%  = PEEKW(inbuf&)                            048
   iHeight% = PEEKW(inbuf& + 2)                        108
   iDepth%  = PEEK(inbuf& + 8)                         251
   iCompr%  = PEEK(inbuf& + 10)                        835
   scrWidth%  = PEEKW(inbuf& + 16)                     034
   scrHeight% = PEEKW(inbuf& + 18)                     220
   iRowBytes% = iWidth% /8                             983
   scrRowBytes% = scrWidth% / 8                        112
   nColors%  = 2^(iDepth%)                             246
   AvailRam& = FRE(-1)                                 070
   NeededRam& =
     ((scrWidth%/8)*scrHeight%*(iDepth%+1))+5000       598
   IF AvailRam& < NeededRam& THEN                      966
      loadError$ = "Rechner-Speicherplatz reicht
       nicht aus."                                     543
      GOTO Lcleanup                                    528
   END IF                                              918
   kk = 1                                              686
   IF scrWidth% > 320 THEN END                         383
   IF scrHeight% > 200  THEN END                       102
   GOSUB GetScrAddrs                                   716
ELSEIF tt$ = "CMAP" THEN  ' Farb-Palette               637
   foundCMAP = 1                                       472
   rLen& = xRead&(fHandle&,cbuf&,icLen&)               042
rurz:                                                  947
   FOR kk = 0 TO nColors% - 1                          855
      red% = PEEK(cbuf&+(kk*3))                        271
      gre% = PEEK(cbuf&+(kk*3)+1)                      062
      blu% = PEEK(cbuf&+(kk*3)+2)                      283
      regTemp% = (red%*16)+(gre%)+(blu%/16)            320
      POKEW(ctab&+(2*kk)),regTemp%                     722
      Farben%(kk,0)=red%/16:IF Farben%(kk,0)>16
       THEN Farben%(kk,0)=16                           672
      Farben%(kk,1)=gre%/16:IF Farben%(kk,1)>16
       THEN Farben%(kk,1)=16                           228
      Farben%(kk,2)=blu%/16:IF Farben%(kk,2)>16
       THEN Farben%(kk,2)=16                           492
   NEXT                                                619
ELSEIF tt$ = "CAMG" THEN                               975
   foundCAMG = 1                                       377
   rLen& = xRead&(fHandle&,inbuf&,icLen&)              731
   camgModes& = PEEKL(inbuf&)                          926
ELSEIF tt$ = "CCRT" THEN                               182
   foundCCRT = 1                                       100
   rLen& = xRead&(fHandle&,inbuf&,icLen&)              582
   ccrtDir%     = PEEKW(inbuf&)                        346
   ccrtStart%   = PEEK(inbuf& + 2)                     099
   ccrtEnd%     = PEEK(inbuf& + 3)                     134
   ccrtSecs&    = PEEKL(inbuf& + 4)                    842
   ccrtMics&    = PEEKL(inbuf& + 8)                    128
ELSEIF tt$ = "BODY" THEN                               198
   foundBODY = 1                                       594
   IF iCompr% = 0 THEN                                 424
      FOR rr = 0 TO iHeight% -1                        484
         FOR pp = 0 TO iDepth% -1                      072
            scrRow& = bPlane&(pp)+(rr*scrRowBytes%)    746
```

```
            rLen& =
              xRead&(fHandle&,scrRow&,iRowBytes%)   480
          NEXT                                      951
        NEXT                                        870
      ELSEIF iCompr% = 1 THEN                       366
        FOR rr = 0 TO iHeight% -1                   026
          FOR pp = 0 TO iDepth% -1                  702
            scrRow& = bPlane&(pp)+(rr*scrRowBytes%) 096
            bCnt% = 0                               853
            WHILE (bCnt% < iRowBytes%)              348
              rLen& = xRead&(fHandle&,inbuf&,1)     890
              inCode% = PEEK(inbuf&)                334
              IF inCode% < 128 THEN                 766
                rLen& = xRead&(fHandle&,scrRow& +
                 bCnt%, inCode%+1)                  309
                bCnt% = bCnt% + inCode% + 1         796
              ELSEIF inCode% > 128 THEN             947
                rLen& = xRead&(fHandle&,inbuf&,1)   893
                inByte% = PEEK(inbuf&)              102
                FOR kk = bCnt% TO bCnt% + 257 -
                 inCode%                            496
                  POKE(scrRow&+kk),inByte%          071
                NEXT                                246
                bCnt% = bCnt% + 257 - inCode%       196
              END IF                                938
            WEND                                    460
          NEXT                                      075
        NEXT                                        850
      ELSE                                          603
        loadError$ = "Unbekanntes
         Komprimierungsverfahren."                  335
        GOTO Lcleanup                               104
      END IF                                        302
ELSE                                                991
    FOR kk = 1 TO icLen&                            354
      rLen& = xRead&(fHandle&,inbuf&,1)             712
    NEXT                                            775
    IF (icLen& OR 1) = icLen& THEN                  391
      rLen& = xRead&(fHandle&,inbuf&,1)             321
    END IF                                          023
END IF                                              165
IF foundBMHD AND foundCMAP AND foundBODY THEN       971
    GOTO GoodLoad                                   933
END IF                                              555
IF rLen& > 0 THEN GOTO ChunkLoop                    382
IF rLen& < 0 THEN                                   463
    loadError$ = "Lesefehler."                      699
    GOTO Lcleanup                                   060
END IF                                              354
IF (foundBMHD=0) OR (foundBODY=0) OR (foundCMAP=0)
 THEN                                               750
    loadError$ = "Wichtige ILBM-Chunks nicht
     gefunden."                                     798
    GOTO Lcleanup                                   275
END IF                                              681
GoodLoad:                                           712
loadError$ = ""                                     203
IF foundCMAP THEN                                   310
    CALL LoadRGB4&(sViewPort&,ctab&,nColors%)       318
END IF                                              956
Lcleanup:                                           474
IF fHandle& <> 0 THEN CALL xClose&(fHandle&)        783
IF mybuf& <> 0 THEN CALL
 FreeMem&(mybuf&,mybufsize&)                        469
RETURN                                              581
Scleanup:                                           574
IF fHandle& <> 0 THEN CALL xClose&(fHandle&)        539
IF mybuf& <> 0 THEN CALL
 FreeMem&(mybuf&,mybufsize&)                        137
RETURN                                              601
GetScrAddrs:                                        497
    sWindow&    = WINDOW(7)                         543
    sScreen&    = PEEKL(sWindow& + 46)              829
    sViewPort& = sScreen& + 44                      195
    sRastPort& = sScreen& + 84                      200
    sColorMap& = PEEKL(sViewPort& + 4)              861
    colorTab&  = PEEKL(sColorMap& + 4)              285
    sBitMap&   = PEEKL(sRastPort& + 4)              853
    scrWidth%  = PEEKW(sScreen& + 12)               759
    scrHeight% = PEEKW(sScreen& + 14)               304
    scrDepth%  = PEEK(sBitMap& + 5)                 907
    nColors%   = 2^scrDepth%                        135
    FOR kk = 0 TO scrDepth% - 1                     224
       bPlane&(kk) = PEEKL(sBitMap&+8+(kk*4))       027
    NEXT                                            034
RETURN                                              514
vorbereitungen:                                     443
  Farben=5 :                                        453
  DIM Farben%(31,2),pointer(31),erlaubt(31)         555
  DIM clip%(1750),tausch%(1750)                     391
  WINDOW 2,"Zaubermeister c 1988 by
   NEUDELSOFT",,16,1                                330
  WINDOW CLOSE 3                                    359
  WINDOW CLOSE 4                                    765
  WINDOW 2                                          223
  PALETTE 0,0,0,0                                   270
  FOR x=1 TO 31                                     442
    PALETTE 32-x,x/32,x/32,x/32                     847
    Farben%(32-x,0)=x/2 : Farben%(32-x,1)=x/2 :
     Farben%(32-x,2)=x/2                            066
  NEXT x                                            737
  MENU 1,0,1,"Projekt"                              483
  MENU 1,1,1,"Bild laden"                           574
  MENU 1,2,1,"Bild speichern"                       079
  MENU 1,3,1,"Farben ändern"                        830
  MENU 1,4,1,"Ende"                                 041
  MENU 1,5,1,"Drucken"                              469
  MENU 1,6,1,"Voreinstellungen"                     826
  MENU 2,0,1,"Farb"                                 141
  MENU 2,1,1,"Heller - Dunkler"                     832
  MENU 2,2,1,"Grautöne"                             634
MENU 2,3,1,"RGB-Töne"                               998
MENU 2,4,1,"Bild verschwimmen lassen"               459
MENU 2,5,1,"Kontraste verstärken"                   139
MENU 2,6,1,"Komplementärfarben"                     993
MENU 2,7,1,"Antik"                                  467
MENU 2,8,1,"Zufallsfarben"                          926
MENU 2,9,1,"RGB - Filter"                           829
MENU 2,10,1,"Farben sortieren (dauert !)"           006
MENU 3,0,1,"Clip"                                   919
MENU 3,1,1,"Markieren"                              195
MENU 3,2,1,"Kopieren"                               495
MENU 3,3,1,"Löschen"                                734
MENU 3,4,1,"Spiegeln"                               914
MENU 3,5,1,"Kippen"                                 975
MENU 3,6,1,"Invertieren"                            517
MENU 3,7,1,"Verändern"                              829
MENU 3,8,1,"Mosaik"                                 482
MENU 4,0,1,"Effekte"                                486
MENU 4,1,1,"Mosaik"                                 241
MENU 4,2,1,"Spiegeln"                               044
MENU 4,3,1,"Kippen"                                 388
MENU 4,4,1,"R.Sei.spieg."                           739
MENU 4,5,1,"L.Sei.spieg."                           229
MENU 4,6,1,"Verschieben"                            511
MENU 4,7,1,"Invertieren"                            327
zeifarbe=1:mosaikx=3:mosaiky=3                      542
MENU ON                                             420
warte:                                              761
MOUSE ON                                            295
COLOR 1,0                                           671
a=0                                                 477
WHILE a=0:a=MENU(0):WEND                            130
b=MENU(1)                                           784
ON a GOTO projekt,farbarbeiten,ausschnitt,effekte   179
effekte:                                            307
ON b GOTO
 mosaik,spiegeln,kippen,rechts,links,verschieben,
invert                                              645
invert:                                             079
amx=mx:amx2=mx2:amy=my:amy2=my2                     897
mx=0:mx2=320:my=0:my2=200                           688
GOTO ainv                                           947
verschieben:                                        907
ax=MOUSE(1):ay=MOUSE(2)                             125
Test=MOUSE(0)                                       152
WHILE MOUSE(0)=0                                    836
 x=MOUSE(1):y=MOUSE(2)                              192
 IF x>ax THEN                                       365
  SCROLL (0,0)-(320,200),1,0                        855
 ELSEIF x<ax THEN                                   635
  SCROLL (0,0)-(320,200),-1,0                       654
 ELSEIF y>ay THEN                                   210
  SCROLL (0,0)-(320,200),0,1                        316
 ELSEIF y<ay THEN                                   030
  SCROLL (0,0)-(320,200),0,-1                       141
 END IF                                             491
 ax=x:ay=y                                          791
WEND                                                393
GOTO warte                                          277
mosaik:                                             267
amx=mx:amx2=mx2:amy=my:amy2=my2                     557
mx=0:mx2=320:my=0:my2=200                           980
GOTO amos                                           925
kippen:                                             134
amx=mx:amx2=mx2:amy=my:amy2=my2                     340
mx=0:mx2=320:my=0:my2=186                           692
GOTO akipp                                          002
spiegeln:                                           555
amx=mx:amx2=mx2:amy=my:amy2=my2                     997
mx=0:mx2=310:my=0:my2=200                           363
GOTO aspi                                           441
rechts:                                             616
c=0                                                 812
FOR b=156 TO 311                                    142
 c=c+1                                              997
 GET (b,0)-(b,200),clip%                            919
 PUT (156-c,0),clip%,PSET                           478
NEXT b                                              159
GOTO warte                                          835
links:                                              986
Test=-1                                             717
FOR b=155 TO 0 STEP -1                              031
 Test=Test+1                                        488
 GET (b,0)-(b,200),clip%                            878
 PUT (156+Test,0),clip%,PSET                        116
NEXT b                                              749
GOTO warte                                          129
amos:                                               431
FOR b=my TO my2 STEP mosaiky                        663
 FOR a=mx TO mx2 STEP mosaikx                       687
  farbe=POINT(a,b)                                  282
  IF farbe>=0 THEN                                  687
   LINE (a,b)-(a+mosaikx-1,b+mosaiky-1),farbe,bf    311
  END IF                                            197
 NEXT a                                             885
NEXT b                                              190
GOSUB Testen                                        793
GOTO warte                                          805
ausschnitt:                                         901
ON b GOTO
 amark,akop,aloe,aspi,akipp,ainv,aver,amos          963
aver:                                               831
f$="Bitte geben Sie Punkt A an !"                   023
CALL
 SetWindowTitles(WINDOW(7),SADD(f$+CHR$(0)),0)      717
Test=MOUSE(0)                                       424
WHILE MOUSE(0)=0:WEND                               296
ox=MOUSE(1):oy=MOUSE(2)                             330
IF POINT(ox,oy)<0 THEN BEEP:GOTO aver               751
```

Fortsetzung auf Seite 133

M2 Library-Collector

Im Vergleich der Modula-2-Compiler im Heft 5/88 schnitt der TDI-Compiler nicht sehr gut ab. Trotz des Ergebnisses ist dieser Compiler für viele Hobbyanwender interessant, da er in der Grundausführung nur halb so teuer ist, wie die beiden Konkurrenten des Vergleichs. Auf die Anwender des TDI-Compilers zielt dieses Programm ab.

Wer schon öfter mit dem TDI-Compiler gearbeitet hat, weiß, wie negativ sich der hohe Kaffeekonsum in den Zwangspausen während des Compilierens und Linkens auf die Nerven auswirkt. Da wohl noch nicht sehr viele AMIGA-Besitzer im Besitz einer Festplatte sind, bleibt nur die RAM-Disk, um an Geschwindigkeit wettzumachen, was der Compiler schluckt. Da aber auch hier oft schon bei einem Megabyte die Grenze erreicht ist, heißt es, sparsam mit den Bytes umzugehen, da neben den Libraries ja auch noch der Compiler (160 K) und diverse andere Dinge in den kostbaren Speicher müssen.
Abhilfe kann dadurch geschafft werden, daß nur die Libraries ins RAM kopiert werden, die auch wirklich nötig sind, um das gewünschte Programm zu compilieren und zu linken. Dabei tritt allerdings schnell das Problem auf, welche Libraries überhaupt benötigt werden. Ein Nachschlagen in den Definitionsmodulen bringt zwar einen Teil, aber lange noch nicht alle verwendeten Module an den Tag. Und selbst wenn durch einen abgeschlossenen Linkvorgang alle verwendeten Libraries auf dem Bildschirm stehen, so ist es immer noch äußerst mühsam, dieses Wissen in ein Batch-File umzusetzen, das diese in die RAM-Disk kopiert.

Collect hält alle Puzzleteile zusammen

An eben dieser Stelle greift das Programm Collect ein. Als Eingabe wird nur benötigt, welches Modula-Programm zu übersetzen ist, wo sich die Libraries (= *.lnk und *.sym) befinden, und wie das zu erzeugende Batch-File heißen soll. Das Programm Collect sucht danach alle (!) Module zusammen und legt ein Batch-File an, das aus der Gesamtmenge der vorhandenen Implementationsmodule diejenigen in die RAM-Disk kopiert, die für die Erzeugung des angegebenen Programms nötig sind. Dieses Batch-File wird dann einfach durch EXECUTE Batch-File aufgerufen und installiert die benötigten Module im Directory RAM:M2. Zusätzlich wird der Zugriff des Compilers und Linkers auf dieses Directory umgelenkt.
Bedienung:
Das Programm wird als „COLLECT“ aus dem CLI heraus aufgerufen. Hinter COLLECT können noch drei Angaben gemacht werden (falls diese nicht gemacht werden, fordert das Programm sie gesondert):

1. Main-File: die vollständige Namensangabe des Programms, das übersetzt werden soll (zum Beispiel ram:collect.mod).
2. Source-Directory: der vollständige Directory-Pfad, der alle fertigen Implementationsmodule enthält (*.lnk und *.sym), die es gibt (zum Beispiel „M2 Amiga 3.01a D1:M2“).
3. Destination-File: der vollständige Name der Batch-Datei, die angelegt werden soll (zum Beispiel „ram:M2Inst.bat“).

Das Programm geht dann alle Module durch und legt eine Liste der irgendwie vom Main-File verwendeten Module an. Diese Liste wird dann in ein Batch-File transformiert, das man direkt mit
EXECUTE aufrufen kann.
Treten während der Programmausführung Fehler auf, so bricht das Programm mit einer Meldung ab. Besonders eilige Zeitgenossen kopieren am besten die mit EXECUTE auf der Ram-Disk installierten Module auf eine gesonderte Diskette. In Zukunft reichen dann nach dem Einlegen dieser Diskette die folgenden Befehle:
MAKEDIR RAM:M2
COPY DF0: ≠? TO RAM:M2 QUIET
ASSIGN M2: RAM:M2
Dieses Verfahren ist schneller als das Batch-File und selbst auf die gesamte Modul-Disk anzuwenden.
Funktion:
Die Module, aus denen Befehle importiert werden können,

sind in Modula-2 in zwei Teilen gespalten. Zum einen sind dies die Module mit dem Namenskürzel *.LNK. Sie enthalten den eigentlichen Programmcode eines Befehls. Diese Module werden beim Linken benötigt. Sie enthalten übrigens auch die Namen aller weiteren *.LNK-Module, die ein einzelnes *.LNK-Modul verwendet.
Zum zweiten gibt es noch Module mit dem Namenskürzel *.SYM. Diese Module enthalten Beschreibungen zu den Befehlen eines Moduls. Diese Module werden beim Compilieren benötigt, um zu kontrollieren, ob der syntaktische Aufruf eines verwendeten Befehls korrekt ist. Um ein Programm nun vollständig zu erzeugen, werden zunächst zum Compilieren alle *.SYM-Module benötigt, die in den IMPORT-Deklarationen Ihres Programms aufgeführt sind. Zum Linken sind dann alle gleichnamigen Module mit der Endung „.LNK" unverzichtbar, sowie die innerhalb dieser Module angegebenen Module. Diese stimmen nicht unbedingt mit den in den Definitionsmodulen angegebenen Import-Listen überein!

Wie man am besten vorgeht:
Um aus diesem Programm den größten Nutzen zu ziehen, empfiehlt es sich, sich zunächst einmal eine ungefähre Vorstellung davon zu machen, welche Module man überhaupt importieren möchte. Je seltener die Aufrufe von COLLECT sind, desto effektiver wird das einmalige Auslesen und Kopieren der Module.
Sollten Sie in einem Programm zwar schon genau wissen, was an Modulen später benötigt wird, aber Sie brauchen einen Teil dieser Module zur Zeit noch nicht, so können Sie dadurch Zeit sparen, daß Sie die betreffenden IMPORT-Befehle in Kommentarklammern setzen. Diese IMPORT-Befehle werden dann zwar vom Compiler ignoriert, bei einem Aufruf von COLLECT aber schon mitberücksichtigt. Dadurch können Sie bei Bedarf, ohne COLLECT neu aufrufen zu müssen, Ihre IMPORT-Liste ausweiten.

(Björn Görke)

Programmname: Module Collector
Sprache: Modula

Source und compilierte Version auf Leserdiskette

ABC Eingabe mit beliebigem Editor (ohne Zeilennummern) Überprüfen mit checksum

```
MODULE Collector;                                       132
                                                        132
(*  Written Nov.1988 by B.Goerke  *)                    333
(*        TDI MODULA  V.3.01a     *)                    512
(*         Feel free to copy      *)                    009
                                                        009
FROM SYSTEM       IMPORT ADR,ADDRESS,BYTE;              925
FROM Streams      IMPORT
 Done,STREAM,Connect,ReadString,WriteString,WriteCha
r,Disconnect;                                           390
FROM DOSFiles     IMPORT
 FileHandle,ModeOldFile,ModeNewFile,Open,Close,Read;    720
FROM CommandLine IMPORT CLStrings,GetCL;                945
FROM Strings      IMPORT
 String,Concat,Pos,Delete,Length;                       479
FROM DOSLibrary  IMPORT DOSBase,DOSName;                016
FROM Libraries   IMPORT OpenLibrary,CloseLibrary;       609
FROM Conversions IMPORT ConvertToString;                879
IMPORT InOut;                                           592
                                                        592
                                                        592
CONST MaxModules = 200;                                 208
      NL         = 12C;                                 812
      CR         = 15C;                                 094
                                                        094
VAR Module  : ARRAY [1..MaxModules] OF String;          997
    sym     : ARRAY [1..MaxModules] OF BOOLEAN;         079
                                                        079
    con,in,out : FileHandle;                            019
    window     : STREAM;                                868
                                                        868
    argv             : ARRAY [0..4] OF CLStrings;       510
    argc             : CARDINAL;                        034
    main,source,dest : CLStrings;                       014
                                                        014
    ok : BOOLEAN;                                       337
                                                        337
    count,pos : CARDINAL;                               683
                                                        683
    buffer    : ARRAY [0..4000] OF BYTE;                270
    buf       : ADDRESS;                                898
    bufsize   : LONGCARD;                               157
                                                        157
PROCEDURE Equal (s1,s2: ARRAY OF CHAR): BOOLEAN;        865
  VAR i,z : CARDINAL;                                   152
  BEGIN                                                 445
    IF Length(s1) # Length(s2) THEN RETURN FALSE
     END;                                               350
    i:=0;                                               593
    z:=Length(s1);                                      656
    WHILE i<z DO                                        179
      IF s1[i] # s2[i] THEN                             267
        RETURN FALSE;                                   465
      ELSE                                              330
        INC (i);                                        533
      END;                                              059
    END;                                                473
    RETURN TRUE;                                        872
  END Equal;                                            078
                                                        078
PROCEDURE exists (name: ARRAY OF CHAR): BOOLEAN;        522
  VAR nr : CARDINAL;                                    386
  BEGIN                                                 703
    nr:=1;                                              358
    WHILE nr<=count DO                                  184
      IF Equal(name,Module[nr]) THEN RETURN TRUE
       END;                                             696
      INC (nr);                                         722
    END;                                                608
    RETURN FALSE;                                       978
  END exists;                                           612
                                                        612
PROCEDURE CollectFromMain (main :ARRAY OF CHAR):
 BOOLEAN;                                               063
  VAR wort,wort2  : String;                             730
      i           : CARDINAL;                           746
  BEGIN                                                 367
    InOut.OpenInputFile (main);                         951
    IF InOut.Done THEN                                  948
     count:=0;                                          819
     REPEAT                                             400
       InOut.ReadString (wort);                         200
       IF (Equal(wort,"FROM")) OR
        (Equal(wort,"IMPORT")) THEN                     702
         InOut.ReadString (wort2);                      456
         IF Equal(wort,"FROM") THEN                     411
           InOut.ReadString (wort)                      824
         END;                                           526
         IF Pos (wort2,";",1,i) THEN                    534
           Delete (wort2,i,1)                           222
         END;                                           812
         IF NOT Equal(wort2,"SYSTEM") THEN              619
           INC (count);                                 170
           Module [count]:= wort2;                      737
           sym    [count]:= TRUE;                       257
           WriteString (window,">uses ");               429
           WriteString (window,Module [count]);         602
           WriteString (window,"[23SPACES]");           176
           WriteChar   (window,CR);                     794
         END;                                           808
        END;                                            254
     UNTIL (Equal(wort,"VAR")) OR
      (Equal(wort,"BEGIN")) OR (NOT InOut.Done);        922
     InOut.CloseInput;                                  036
     RETURN TRUE;                                       067
    ELSE                                                452
     WriteString (window,"Can't open client module
      !");                                              087
     WriteChar   (window,NL);                           786
     RETURN FALSE;                                      420
```

```
    END;                                                          314
  END CollectFromMain;                                            303
                                                                  303
PROCEDURE CollectFromSource (source: ARRAY OF CHAR):
 BOOLEAN;                                                         622
  VAR wort        : String;                                       987
      name,sl,def: String;                                        114
      i           : CARDINAL;                                     546
      by          : LONGINT;                                      781
      fh          : FileHandle;                                   748
      z,p         : POINTER TO CHAR;                              229
      ch          : CHAR;                                         623
  BEGIN                                                           728
    pos:=1;                                                       349
    WHILE pos<=count DO                                           037
     def:=".lnk";                                                 033
     IF source [Length(source)-1] = ":" THEN sl:=""
      ELSE sl:="/" END;                                           679
     Concat (source,sl,name);                                     704
     Concat (name,Module[pos],wort);                              609
     Concat (wort,def,name);                                      265
     fh:=Open (name,ModeOldFile);                                 517

     IF fh # 0 THEN                                               928
       by := Read (fh,buf,bufsize);                               829
       p  := buf;                                                 068
       (*$T- *)                                                   322
       INC (p,42);                                                782
       ch := p^;                                                  913
       WHILE ch # 0C DO                                           170
         z := p;                                                  602
         i := 0;                                                  809
         REPEAT                                                   574
           ch := z^;                                              435
           wort[i] := ch;                                         465
           INC (i);                                               892
           INC (z,1);                                             309
         UNTIL ch = 0C;                                           379
         IF NOT exists (wort) THEN                                631
           INC (count);                                           222
           Module [count] := wort;                                239
           sym    [count] := FALSE;                               822
           WriteString (window,">uses ");                         218
           WriteString (window,wort);                             388
           WriteString (window,"[32SPACES]");                     026
           WriteChar   (window,CR);                               308
         END;                                                     458
         INC (p,36);                                              325
         ch := p^;                                                468
       END;                                                       130
       (*$T+ *)                                                   022
       Close (fh);                                                878
     ELSE                                                         543
      WriteString (window,"Can't open ");                         868
      WriteString (window,name);                                  067
      WriteChar   (window,NL);                                    902
      RETURN FALSE;                                               072
     END;                                                         030
     INC (pos);                                                   358
    END;                                                          052
    RETURN TRUE;                                                  851
  END CollectFromSource;                                          168
                                                                  168
PROCEDURE Save (dest: ARRAY OF CHAR): BOOLEAN;                    510
  VAR f    : FileHandle;                                          906
      Out  : STREAM;                                              839
      nr   : CARDINAL;                                            918
  BEGIN                                                           027
    f := Open (dest,ModeNewFile);                                 197
    IF f = 0 THEN RETURN FALSE END;                               071
    Connect (Out,f);                                              199
    IF count = 0 THEN RETURN FALSE END;                           076
    nr := 1;                                                      871
    WriteString (Out,"; batch-file installs
     M2-Libraries for module ");                                  815
    WriteString (Out,main);                                       054
    WriteChar   (Out,NL);                                         561
    WriteString (Out,"FAILAT 60");                                546
    WriteChar   (Out,NL);                                         885
    WriteString (Out,"MAKEDIR RAM:M2");                           129
    WriteChar   (Out,NL);                                         920
    WriteString (Out,"ASSIGN M2: RAM:M2");                        969
    WriteChar   (Out,NL);                                         616
    WriteString (Out,"COPY C/COPY TO RAM:");                      627
    WriteChar   (Out,NL);                                         010
    WriteChar   (Out,NL);                                         405
    REPEAT                                                        610
      WriteString (Out,"RAM:COPY ");                              025
      WriteChar   (Out,42C);                                      311

      WriteString (Out,source);                                   526
      WriteString (Out,"/");                                      579
      WriteString (Out,Module[nr]);                               251
      WriteString (Out,".LNK");                                   792
      WriteChar   (Out,42C);                                      126
      WriteString (Out," TO RAM:M2");                             592
      WriteChar   (Out,NL);                                       979
      IF sym [nr] THEN                                            890
        WriteString (Out,"RAM:COPY ");                            833
        WriteChar   (Out,42C);                                    247
        WriteString (Out,source);                                 694
        WriteString (Out,"/");                                    403
        WriteString (Out,Module[nr]);                             755
        WriteString (Out,".SYM");                                 672
        WriteChar   (Out,42C);                                    622
        WriteString (Out," TO RAM:M2");                           000
        WriteChar   (Out,NL);                                     811
      END;                                                        969
      INC (nr);                                                   103
    UNTIL nr > count;                                             101
    Disconnect (Out);                                             275
    Close (f);                                                    219
    RETURN TRUE;                                                  482
  END Save;                                                       507
                                                                  507
                                                                  507
BEGIN                                                             180
  (* Open DOS-Library for Streams and InOut *)                    287
  DOSBase := OpenLibrary (DOSName,0);                             481
                                                                  481
  (* Get command-line *)                                          730
  ok := GetCL (argc,argv);                                        755
                                                                  755
  (* Open console window *)                                       369
  con := Open("*",ModeOldFile);                                   124
  IF con = 0 THEN RETURN END;                                     001
  Connect (window,con);                                           215
                                                                  215
  WriteString (window,"M2 Library-Collector V1.0 (c)
   1988 B.Goerke");                                               013
  WriteChar   (window,NL);                                        792
  WriteString
   (window,"-----------------------------------------
  ---");                                                          973
  WriteChar   (window,NL);                                        216
                                                                  216
  main   := argv [0];        (* source-file        *)             830
  source := argv [1];        (* modules' directory *)             480
[2SPACES]dest[3SPACES]:= argv [2];[6SPACES](* name
 of batch-file *)                                                 706
                                                                  706
[2SPACES]IF main [0] = "?" THEN                                   413
[4SPACES]WriteString (window,"[3SPACES]Usage:
 Modulename + DefSource-directory +
 Destination-file");                                              892
[4SPACES]WriteChar[3SPACES](window,NL);                           991
[4SPACES]RETURN                                                   007
[2SPACES]END;                                                     061
                                                                  061
[2SPACES](* Prepare buffer for Read *)                            026
[2SPACES]buf[5SPACES]:= ADR(buffer);                              943
[2SPACES]bufsize := 4000;                                         941
                                                                  941
[2SPACES]IF argc < 1 THEN                                         797
[4SPACES]WriteString (window,"Main-module[5SPACES]:
 ");                                                              781
[4SPACES]ReadString[2SPACES](window,main);                        761
[2SPACES]END;                                                     263
[2SPACES]IF argc < 2 THEN                                         100
[4SPACES]WriteString (window,"Source-directory: ");               699
[4SPACES]ReadString[2SPACES](window,source);                      567
[2SPACES]END;                                                     213
[2SPACES]IF argc < 3 THEN                                         319
[4SPACES]WriteString (window,"Destination file: ");               017
[4SPACES]ReadString[2SPACES](window,dest);                        804
[2SPACES]END;                                                     538
                                                                  538
[2SPACES]IF CollectFromMain (main) THEN                           549
[4SPACES]IF CollectFromSource (source) THEN                       064
[6SPACES]WriteChar (window,NL);                                   987
[6SPACES]IF Save (dest) THEN                                      021
[8SPACES]WriteString (window,"O.k.");                             465
[8SPACES]WriteChar[3SPACES](window,NL);                           012
[6SPACES]END;                                                     002
[4SPACES]END;                                                     280
[2SPACES]ELSE                                                     769
[4SPACES]WriteString (window,"Seems to be an error
 ....");                                                          291
[4SPACES]WriteChar[3SPACES](window,NL);                           302
[2SPACES]END;                                                     852
                                                                  852
[2SPACES]ConvertToString
 (LONGCARD(count),10,FALSE,main,ok);                              822
[2SPACES]IF ok THEN                                               898
[4SPACES]WriteString (window,main);                               049
[4SPACES]WriteString (window," modules collected.");              937
[4SPACES]WriteChar[3SPACES](window,NL);                           828
[2SPACES]END;                                                     882
                                                                  882
[2SPACES](* Say good-bye to DOS-Library *)                        757
[2SPACES]Disconnect (window);                                     827
[2SPACES]CloseLibrary (DOSBase);                                  695
                                                                  695
END Collector.                                                    591
```

checksum und nanomon sind Eingabehilfen für AmigaWELT-Listings. Sie befinden sich auf jeder Leserdiskette.

Arbeitserleichterung für geplagte TDI-User verspricht dieses Listing.

Fehlerfrei abtippen mit checksum!

Viel Arbeit bei der Fehlersuche nach dem Eintippen von Listings aus AmigaWelt erspart man sich mit „checksum“. Dazu gibt man die Listings (außer Hexdumps) mit einem beliebigen ASCII-Editor (zum Beispiel Ed) ein und läßt sie von „checksum“ auf Fehler überprüfen.

Wenn ein AmigaBASIC-Programm ohne Berücksichtigung der Groß-/Kleinschreibung eingegeben wurde, so muß es von AmigaBASIC aus mit ‹LOAD „Programmname“› eingeladen werden und mit ‹SAVE „Programmname“ ,a› abgespeichert werden. Bei der Eingabe können Kommentare inclusive der Begrenzer ‹REM› und ‹’› (Alt und ä gemeinsam) in Basic sowie ‹/*. . .*/› in C weggelassen werden. Das war in der ersten Version von „checksum“ noch nicht möglich, daher muß bei allen Programmen ab Ausgabe 4/88 der neue „checksum“ verwendet werden. Ebenso können Leerzeilen und zur Übersichtlichkeit und Strukturierung eingesetzte Leerzeichen (Freiräume außerhalb von Strings) weggelassen werden. Ein Leerzeichen innerhalb eines Strings findet in der Prüfsumme Berücksichtigung. Zwei oder mehr Leerzeichen innerhalb eines Strings werden im ausgedruckten Listing durch [xSPACES] dargestellt. „x“ steht für die Zahl der Leerzeichen.

Zeilennummern werden nicht mit eingegeben

Die aus drei Ziffern bestehende Prüfsumme befindet sich am Ende einer Zeile und kann unmittelbar hinter oder durch beliebig viele Leerzeichen getrennt vom Programmtext eingegeben werden. Wichtig ist, daß die Zeilennummern nicht mit eingegeben werden, sie dienen nur zur Übersicht. Steht bei einer Zeile keine Zeilennummer, so gehört diese Zeile noch zu der vorherigen und muß unmittelbar hinter dieser eingegeben werden — aus den beiden Zeilen im Listing wird eine Zeile im Editor. Nachdem das komplette Listing mit dem ASCII-Editor eingegeben und gespeichert wurde, wird das Programm „checksum“ gestartet. Die AmigaBASIC-Version wird im AmigaBASIC geladen und gestartet, die benötigten Eingaben werden anschließend abgefragt.

Auf jeder Leserdiskette befindet sich eine compilierte C-Version. Sie wird vom CLI aus mit

checksum [—l‹*sprache*›] [—o‹*ausgabe*›] [—l‹*start*›] [—s‹*schritt*›] ‹quellfile›

aufgerufen. Dabei bedeuten die [ ]-Klammern, daß die Eingabe weggelassen werden kann. In diesem Fall wird ein Default-Wert (Voreinstellung) angenommen.
‹sprache› entweder C, ASSEM oder BASIC (Batchdateien werden als Assembler abgedruckt)

‹ausgabe› Name des Ausgabefiles (ohne Prüfsummen)
‹start› Tabulatorstart (erste Position, 1 bis 64)
‹schritt› Tabulatorschrittweite (Abstand, 1 bis 32)
‹quellfile› Name des zu überprüfenden Files

Die Reihenfolge der Parameter spielt keine Rolle. Die Defaultwerte sind C für ‹sprache› (bei BASIC-Version BASIC) , CLI-Fenster für ‹ausgabe›, 1 für ‹start›, 8 für ‹schritt› und für ‹quellfile› gibt es keinen Defaultwert — es muß auf jeden Fall ein Quellfile angegeben werden. Wurde ein Assembler-Quelltext unter dem Namen „quell.sum“ gespeichert, so wird durch ‹checksum —lassem —oquell.asm quell.sum› das eigentliche Quellfile mit dem Namen „quell.asm“ erzeugt.

Prüfsummen werden vom Programm automatisch entfernt

Dabei werden die Prüfsummen entfernt und das Programm auf Fehler untersucht. Es gibt folgende Fehlermeldungen:

„no checksum“, die Zeile enthält keine Prüfsumme
„checksum error“, fehlerhafte Zeile
„unexpected end of file“, die Überprüfung traf innerhalb von Kommentaren oder Anführungszeichen auf das Dateiende
„too many tabs“, die Zeile enthält zu viele Tabulatoren, so daß der interne Puffer voll ist

Wenn Fehler aufgetreten sind, wird das Programm wieder in den ASCII-Editor geladen und die Fehler ausgebessert. Danach wird es nochmals mit „checksum“ überprüft. Dies wird solange wiederholt, bis keine Fehler mehr auftreten. Dann existiert das lauffähige Programm unter dem bei *‹ausgabe›* angegebenen Namen. Wenn die Ausgabe auf den Bildschirm erfolgte, muß das Programm nochmals mit Angabe eines Dateinamens bei *‹ausgabe›* von checksum bearbeitet werden.

(Text von T. Borris, Programme von R. Babel) ►

Programmname: checksum
Sprache: BASIC

Auf der Leserdiskette befindet sich nur das lauffähige C-Compilat!

ABC Eingabe mit beliebigem Editor (ohne Zeilennummern) Überprüfen mit checksum (alte Version)

```
REM                                                      496
REM "CheckSum.bas - Update 08-Apr-1988 für AmigaBASIC    672
REM "Copyright (C) 1988 CW-Publikationen, München        272
REM "all rights reserved - alle Rechte vorbehalten       064
REM                                                      840
                                                         840
DEFINT a-z                                               022
                                                         022
' "Bereich für Prüfsumme (0 .. SUMRANGE-1)               325
                                                         325
SUMWIDTH = 3                                             214
SUMRANGE = 1000                                          542
                                                         542
' "unterstützte Sprachen (lmode)                         941
                                                         941
L.C     = 1                                              396
L.BASIC = 2                                              551
L.ASSEM = 3                                              622
                                                         622
' "Bitpositionen (qmode)                                 981
                                                         981
Q.ENCLOSED = &O1                                         756
                                                         756

Q.IGNORE = &O2                                           191
                                                         191
Q.DISCARD  = &O14                                        083
Q.DISCARD1 = &O4                                         267
Q.DISCARD2 = &O10                                        821
Q.DISCARD3 = &O14                                        808
                                                         808
Q.C.MAINMODES = &O60                                     177
Q.C.SINGLE    = &O20                                     027
Q.C.DOUBLE    = &O40                                     788
Q.C.COMMENT   = &O60                                     847
Q.C.NONWHITE  = &O100                                    794
                                                         794
Q.BASIC.DOUBLE    = &O20                                 988
Q.BASIC.DATA[5SPACES]= &O40                              151
Q.BASIC.REM       = &O100                                322
Q.BASIC.NONWHITE = &O200                                 769
Q.BASIC.PRECSYM  = &O400                                 657
                                                         657
Q.ASSEM.MAINMODES = &O60                                 040
Q.ASSEM.SINGLE    = &O20                                 092
Q.ASSEM.DOUBLE    = &O40                                 623
Q.ASSEM.COMMENT   = &O60                                 888
Q.ASSEM.NONWHITE  = &O100                                297
Q.ASSEM.NOTFIRST  = &O200                                176
                                                         176
' "Defaults etc.                                         279
                                                         279
MINSTART      = 1                                        731
MAXSTART      = 64                                       834
DEFAULT.START = 1                                        889
                                                         889
MINSTEP      = 1                                         135
MAXSTEP      = 32                                        279
DEFAULT.STEP = 8                                         847
                                                         847
' "Parameter auswerten                                   810
                                                         810
loop1:                                                   439
INPUT"Sprache (Default = BASIC)";a$:a$=UCASE$(a$)        449
IF a$="C" THEN                                           448
  lmode=L.C                                              459
ELSEIF a$="" OR a$="BASIC" THEN                          113
  lmode=L.BASIC                                          599
ELSEIF a$="ASSEM" THEN                                   925
  lmode=L.ASSEM                                          914
ELSE                                                     643
  GOTO loop1                                             487
END IF                                                   989
                                                         989
loop2:                                                   311
PRINT"Tabulator-Start (Default ="DEFAULT.START")";       160
INPUT tabstart                                           841
IF tabstart=0 THEN                                       133
  tabstart=DEFAULT.START                                 820
ELSEIF tabstart<MINSTART OR tabstart>MAXSTART THEN       642
  GOTO loop2                                             007
END IF                                                   253
                                                         253
loop3:                                                   332
PRINT"Tabulator-Schrittweite (Default ="DEFAULT.STEP")"; 156
INPUT tabstep                                            063
IF tabstep=0 THEN                                        701
  tabstep=DEFAULT.STEP                                   552
ELSEIF tabstep<MINSTEP OR tabstep>MAXSTEP THEN           558
  GOTO loop3                                             532
END IF                                                   610
                                                         610
INPUT"Quelldatei";srcname$                               092
                                                         092
INPUT"Zieldatei (Default = Bildschirm)";dstname$         382
IF dstname$="" THEN dstname$="SCRN:                      400
                                                         400
' "Files öffnen, Summen prüfen und Quelltext erzeugen    143
                                                         143
OPEN srcname$ FOR INPUT AS 1                             182
OPEN dstname$ FOR OUTPUT AS 2                            914
                                                         914
qmode    = 0                                             969
checksum = 0                                             617
nline    = 0                                             408
                                                         408
WHILE NOT EOF(1)                                         586
  nline=nline+1                                          943
  LINE INPUT#1,buffer$                                   889
                                                         889
  ' "Tabulatoren durch Leerzeichen ersetzen              356
                                                         356
  col=1                                                  408
  FOR i=1 TO LEN(buffer$)                                871
    IF MID$(buffer$,col,1)=CHR$(9) THEN                  785
      IF col<tabstart THEN                               806
        ins=tabstart-col                                 149
      ELSE                                               590
        ins=tabstep-(col-tabstart) MOD tabstep-1         047
      END IF                                             509
      buffer$=LEFT$(buffer$,col-1)+STRING$(ins+1,"
        ")+MID$(buffer$,col+1)                           573
      col=col+ins                                        711
    END IF                                               469
    col=col+1                                            270
  NEXT                                                   685
                                                         685
  GOSUB rmtrlspc                                         636
                                                         636
  IF LEN(buffer$)<SUMWIDTH GOTO nosum                    845
                                                         845
  givensum=0                                             916
  i=3                                                    473
  loop:                                                  741
    digit=ASC(LEFT$(RIGHT$(buffer$,i),1))-48             463
    IF digit<0 OR digit>9 GOTO nosum                     407
    givensum=givensum*10+digit                           515
  i=i-1:IF i>0 GOTO loop ELSE skip                       555
                                                         555
  nosum:                                                 239
  givensum=-1                                            816
  PRINT "line "nline": no checksum                       747
                                                         747
  skip:                                                  284
  IF givensum<>-1 THEN                                   828
    buffer$=LEFT$(buffer$,LEN(buffer$)-SUMWIDTH)         921
    GOSUB rmtrlspc                                       624
  END IF                                                 246
                                                         246
  buffer$=buffer$+CHR$(10)                               457
  PRINT#2,buffer$;                                       960
                                                         960
  "irrelevante Zeichen entfernen und Prüfsumme bilden    231
                                                         231
  FOR i=1 TO LEN(buffer$)                                314
    c$=MID$(buffer$,i,1)                                 781
                                                         781
    IF qmode AND Q.DISCARD THEN ' "Ausgleich 'look-ahead'
      o. ä.                                              408
      qmode=qmode AND NOT Q.DISCARD OR (qmode AND
      Q.DISCARD)-Q.DISCARD1                              845
    ELSE                                                 110
      IF lmode=L.C THEN                                  191
        IF (qmode AND Q.C.MAINMODES)=Q.C.SINGLE THEN     491
          IF c$="'" THEN                                 664
            qmode=qmode AND NOT (Q.ENCLOSED OR
             Q.C.MAINMODES)                              340
          ELSE                                           869
            qmode=qmode OR Q.ENCLOSED                    279
            IF c$="\" THEN                               841
              qmode=qmode OR Q.DISCARD1                  729
            END IF                                       727
          END IF                                         333
        ELSEIF (qmode AND Q.C.MAINMODES)=Q.C.DOUBLE THEN 439
          IF c$=CHR$(34) THEN                            938
            qmode=qmode AND NOT (Q.ENCLOSED OR
             Q.C.MAINMODES)                              678
          ELSE                                           215
            qmode=qmode OR Q.ENCLOSED                    929
            IF c$="\" THEN                               043
              qmode=qmode OR Q.DISCARD1                  019
            END IF                                       425
          END IF                                         783
        ELSEIF (qmode AND Q.C.MAINMODES)=Q.C.COMMENT THEN 849
          IF MID$(buffer$,i,2)="*/" THEN                 825
            qmode=qmode AND NOT Q.C.MAINMODES OR
             Q.DISCARD1 OR Q.IGNORE                      990
          ELSEIF c$=CHR$(10) THEN                        180
            qmode=qmode AND NOT Q.IGNORE                 760
          ELSE                                           353
            qmode=qmode OR Q.IGNORE                      526
          END IF                                         524
        ELSE                                             925
          IF c$=" " THEN                                 387
            IF qmode AND Q.C.NONWHITE THEN               178
              qmode=qmode AND NOT Q.C.NONWHITE           015
            ELSE                                         080
              qmode=qmode OR Q.IGNORE                    561
            END IF                                       367
          ELSE                                           968
```

```
            qmode=qmode AND NOT Q.IGNORE OR Q.C.NONWHITE  705
                                                          705
            IF c$=CHR$(10) THEN                           334
              qmode=qmode AND NOT (Q.IGNORE OR
                Q.C.NONWHITE)                             240
            ELSEIF c$="'" THEN                            510
              qmode=qmode OR Q.C.SINGLE                   434
            ELSEIF c$=CHR$(34) THEN                       030
              qmode=qmode OR Q.C.DOUBLE                   011
            ELSEIF MID$(buffer$,i,2)="/*" THEN            752
              qmode=qmode OR Q.C.COMMENT OR Q.DISCARD1 OR
                Q.IGNORE                                  284
            END IF                                        834
          END IF                                          160
        END IF                                            422
      ELSEIF lmode=L.BASIC THEN                           751
        IF c$=CHR$(10) THEN                               428
          qmode=0                                         575
        ELSEIF (qmode AND Q.BASIC.REM)=0 THEN             043
          IF c$=CHR$(34) THEN                             582
            qmode=qmode XOR Q.BASIC.DOUBLE                914
            IF (qmode AND Q.BASIC.DOUBLE)=0 THEN          533
              IF (qmode AND Q.BASIC.DATA)=0 THEN          467
                qmode=qmode AND NOT Q.ENCLOSED            500
              END IF                                      834
            END IF                                        160
            GOTO quit                                     964
          END IF                                          658
          IF qmode AND Q.BASIC.DOUBLE THEN                159
            qmode=qmode OR Q.ENCLOSED                     945
          END IF                                          727
          IF qmode AND Q.BASIC.DATA THEN                  403
            IF (qmode AND Q.BASIC.DOUBLE)=0 AND c$=":"
              THEN                                        375
              qmode=qmode AND NOT (Q.ENCLOSED OR
                Q.BASIC.DATA)                             951
              GOTO quit                                   335
            END IF                                        541
          ELSE                                            086
            qmode=qmode OR Q.ENCLOSED                     028
          END IF                                          578
          IF (qmode AND (Q.BASIC.DOUBLE OR
          Q.BASIC.DATA))=0 THEN                           140
            IF c$=" " THEN                                794
              IF qmode AND Q.BASIC.NONWHITE THEN          952
                qmode=qmode AND NOT Q.BASIC.NONWHITE      104

              ELSE                                        441
                qmode=qmode OR Q.IGNORE                   710
              END IF                                      796
            ELSE                                          453
              qmode=qmode AND NOT Q.IGNORE OR
                Q.BASIC.NONWHITE                          401
              IF c$="'" THEN                              102
                qmode=qmode OR Q.BASIC.REM OR Q.IGNORE    675
              ELSEIF (qmode AND Q.BASIC.PRECSYM)=0 THEN   383
                CALL isbasicsym(MID$(buffer$,i+3,1),flag) 118
                IF UCASE$(MID$(buffer$,i,3))="REM" AND
                  NOT flag THEN                           186
                  qmode=qmode OR Q.BASIC.REM OR
                    Q.DISCARD2 OR Q.IGNORE                383
                ELSE                                      192
                  CALL
                    isbasicsym(MID$(buffer$,i+4,1),flag)  144
                  IF UCASE$(MID$(buffer$,i,4))="DATA" AND
                    NOT flag THEN                         615
                    qmode=qmode OR Q.BASIC.DATA OR
                      Q.DISCARD3                          823
                  END IF                                  709
                END IF                                    075
              END IF                                      545
            END IF' "else c == ' '                        858
          END IF' "if no modes on                         447
        quit:                                             772
        qmode=qmode AND NOT Q.BASIC.PRECSYM               795
        CALL isbasicsym(c$,flag)                          612
        IF flag THEN qmode=qmode OR Q.BASIC.PRECSYM       944
        END IF                                            054
      ELSEIF lmode=L.ASSEM THEN                           930
        IF c$=CHR$(10) THEN                               411
          qmode=0                                         114
        ELSE                                              379
          IF (qmode AND Q.ASSEM.MAINMODES)=Q.ASSEM.SINGLE
           THEN                                           187
            IF c$="'" THEN                                232
              IF MID$(buffer$,i+1,1)="'" THEN             733
                qmode=qmode OR Q.DISCARD1 OR Q.ENCLOSED   502
              ELSE                                        319
                qmode=qmode AND NOT (Q.ENCLOSED OR
                  Q.ASSEM.MAINMODES)                      293
              END IF                                      939
            ELSE                                          964
              qmode=qmode OR Q.ENCLOSED                   266
            END IF                                        576
          ELSEIF (qmode AND
          Q.ASSEM.MAINMODES)=Q.ASSEM.DOUBLE THEN          614
            IF c$=CHR$(34) THEN                           093
              qmode=qmode AND NOT (Q.ENCLOSED OR
                Q.ASSEM.MAINMODES)                        931
            ELSE                                          804
              qmode=qmode OR Q.ENCLOSED                   210
            END IF                                        456
          ELSEIF (qmode AND
          Q.ASSEM.MAINMODES)<>Q.ASSEM.COMMENT THEN        731
            IF c$=" " THEN                                785
              IF qmode AND Q.ASSEM.NONWHITE THEN          022
                qmode=qmode AND NOT Q.ASSEM.NONWHITE      209
              ELSE                                        074
                qmode=qmode OR Q.IGNORE                   571
              END IF                                      145
            ELSE                                          034
              qmode=qmode AND NOT Q.IGNORE OR
                Q.ASSEM.NONWHITE                          961
                                                          961
              IF c$="'" THEN                              790
                qmode=qmode OR Q.ASSEM.SINGLE             384
              ELSEIF c$=CHR$(34) THEN                     972
                qmode=qmode OR Q.ASSEM.DOUBLE             383
              ELSEIF c$="*" AND (qmode AND
              Q.ASSEM.NOTFIRST)=0 OR c$=";"               814
 THEN                                                     841
                qmode=qmode OR Q.ASSEM.COMMENT OR
                  Q.IGNORE                                428
              END IF                                      850
            END IF                                        072
          END IF                                          526
          qmode=qmode OR Q.ASSEM.NOTFIRST                 011
        END IF                                            993
      END IF' "switch lmode                               002
    END IF' "else qmode and Q.DISCARD                     007
                                                          007
    IF (qmode AND Q.IGNORE)=0 and c$<>chr$(10) then       634
      checksum=checksum*13+ASC(c$) AND &H7FF              429
    END IF                                                523
  NEXT                                                    274
                                                          274
  ' "won't handle trailing spaces                         553
                                                          553
  checksum=checksum MOD SUMRANGE                          754
                                                          754
  IF givensum<>-1 AND checksum<>givensum THEN             176
    PRINT "line "nline": checksum error                   604
    checksum=givensum                                     606
  END IF                                                  988
WEND                                                      006
                                                          006
' "alle Modi müssen abgeschaltet sein!                    117
                                                          117
IF qmode<>0 THEN PRINT "line "nline": unexpected end of
  file                                                    659
                                                          659
CLOSE 2,1                                                 600
                                                          600
END                                                       791
                                                          791
' "Leerzeichen am Zeilenende entfernen                    082
                                                          082
rmtrlspc:                                                 679

                                                          679
WHILE RIGHT$(buffer$,1)=" "                               035
  buffer$=LEFT$(buffer$,LEN(buffer$)-1)                   744
WEND                                                      466
                                                          466
RETURN                                                    562
                                                          562
' "Test auf erlaubtes Zeichen für BASIC-Symbol            201
                                                          201
SUB isbasicsym(c$,flag) STATIC                            816
                                                          816
  flag=1                                                  960
                                                          960
  IF UCASE$(c$)<"A" OR UCASE$(c$)>"Z" THEN IF c$<"0" OR
    c$>"9" THEN IF                                        865
c$<>"." THEN flag=0                                       014
                                                          014
END SUB                                                   059
```

Für BASIC-Fans ist diese Version von checksum gedacht.

Vom Eiffelturm zum Fudschijama

Mit „Jet" und „Flightsimulator" nach Japan, Westeuropa und Amerika — Scenery Disks führen in die weite Welt.

Daß Sublogic's Mitarbeiter selber Piloten sind oder sich zumindest in der Fliegerei auskennen, zeigt sich an den Landschaftsdisketten. Man hatte reichlich Flugkarten zur Verfügung und konnte Landschaften maßstabgetreu abbilden.

Für den Amiga gibt es vier Scenery Disks. Weitere sind in Vorbereitung.

In der Verpackung befinden sich eine Diskette, Flugkarten mit Airport-Koordinaten, eine englische Anleitung (Im Großen Amiga Spielebuch von technicSupport wird im Artikel „In 10 Flugstunden zum Pilotenschein" eine deutsche Anleitung zur Western European Tour gegeben) und eine Liste der Koordinaten und Frequenzen aller Funkfeuer. Erhältlich sind:

— „Western European Tour":
Südengland, Nordfrankreich und Süddeutschland

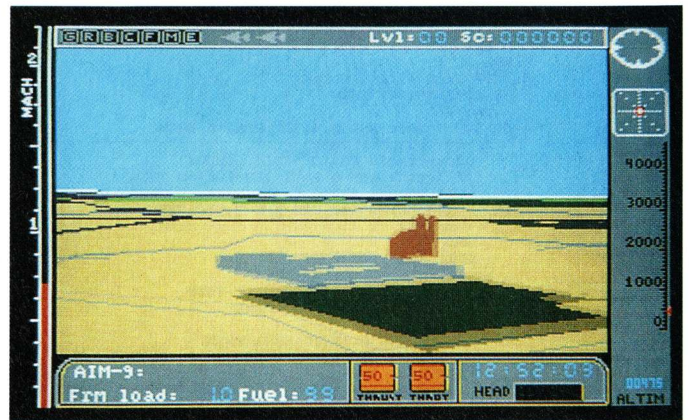
— „Japan Scenery Disk":
Tokyo, Osaka

— „Scenery Disk 7":
Washington, Charlotte, Jacksonville, Miami

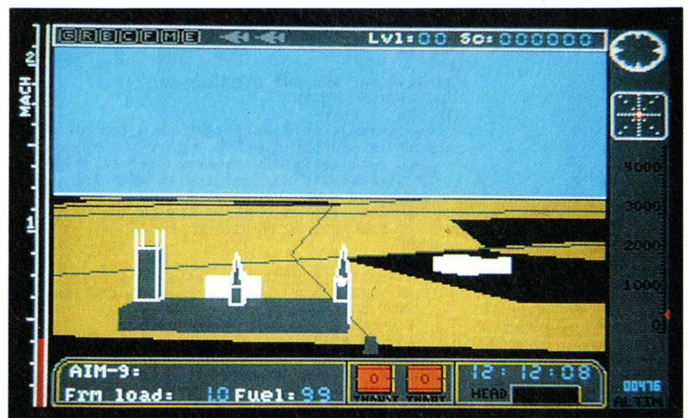
— „Scenery Disk 11":
Lake Huron, Detroit

Sublogic's Landschaftsdisketten sind zu „Jet" und „Flightsimulator I und II" kompatibel. Unsere Screenshots stammen von einem Rundflug mit dem „Jet".

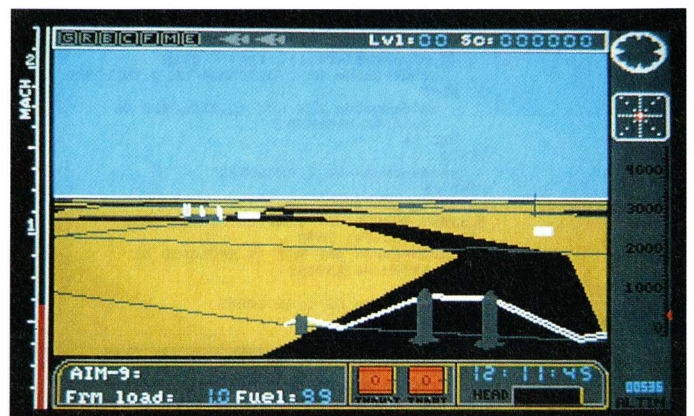
Jede Scenery Disk kostet 50 Mark. Vertreiber sind Rushware, technicSupport, GTI.



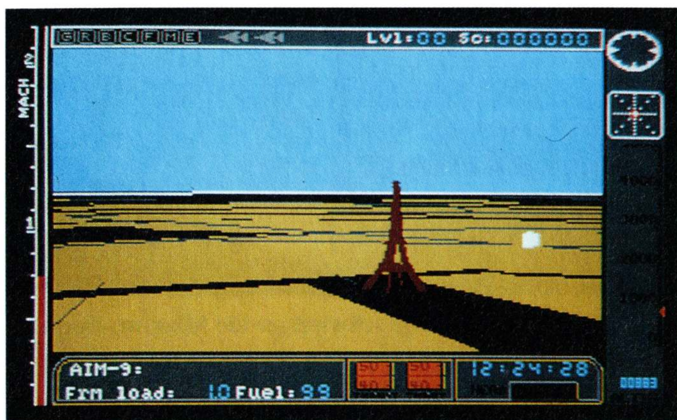
Münchens Frauentürme grüßen den Luftreisenden aus der Ferne. Am Horizont sind schneebedeckte Gipfel angedeutet.



Die Houses of Parliament sind weithin sichtbar. Der kleinere Turm ist der Big Ben mit der berühmten Stundenglocke.



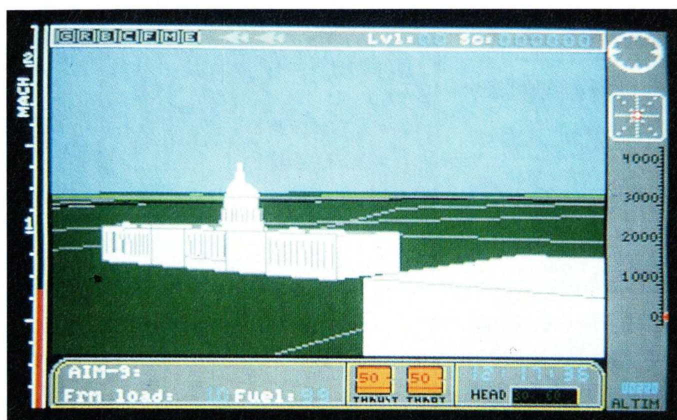
Die Towerbridge überspannt die Themse. Die gigantische Zugbrücke ist nach Londons ältestem Gebäude benannt.



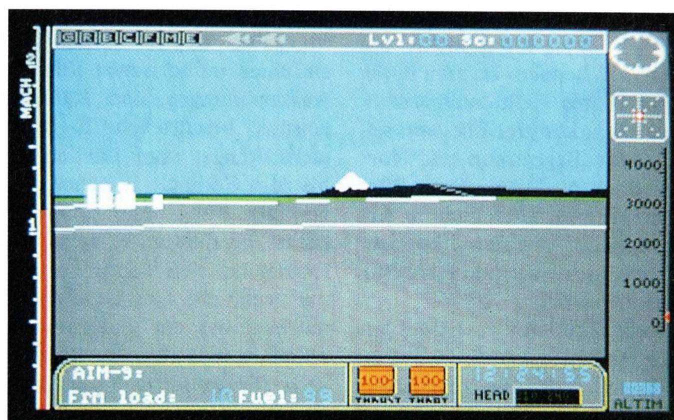
320 Meter ragt der Eiffelturm in den Himmel. Das gigantische Bauwerk dominiert die Skyline von Paris.



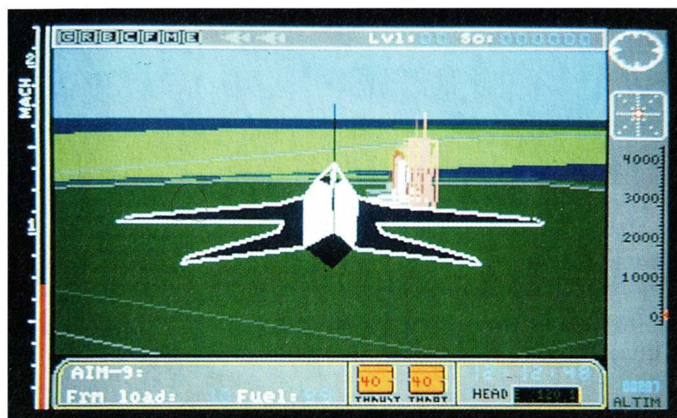
In Toronto, der Hauptstadt der kanadischen Provinz Ontario, steht der höchste Fernsehturm der Welt.



In Washington D.C., am Ufer des Potomac, thront das Weiße Haus. Von dort aus ist es nicht weit bis zum Pentagon.



Zur Linken ragen Tokios Wolkenkratzer in den Himmel. Am Horizont sieht man den heiligen Berg Fudschi.



In Cape Canaveral im Staat Miami wird gerade die Raumfähre Challenger startklar gemacht.



Daß die Kraterwände des Fudschijama nach innen abbrechen und mit Beton abgestützt worden sind, ist nicht sichtbar.

F-16 Falcon

Rasante Mischung

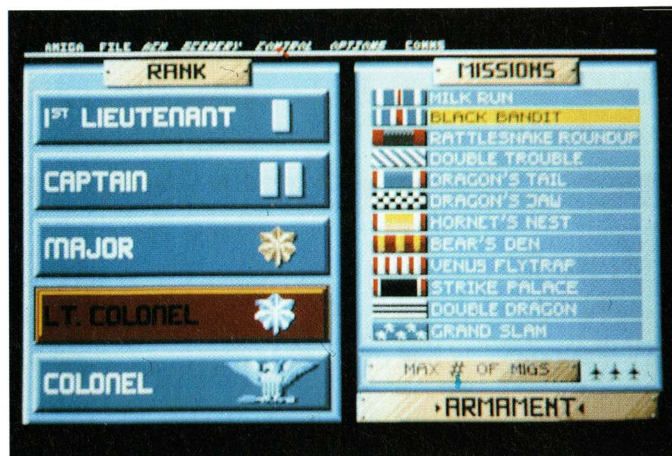
Was kommt dabei heraus, wenn man „Jet“, „Flightsimulator III“ und „Interceptor“ kreuzt? Das Ergebnis heißt „F-16 Falcon“ und läßt auf Gewichtiges schließen. Ursache dafür ist das knapp 150-seitige Handbuch. Beim oberflächlichen Durchblättern verschlägt es einem den Atem, denn das Buch ist in Englisch geschrieben und strotzt von Fachausdrükken. Nun das Buch zur Seite gelegt, die Keyboard-Command-Karte aufgestellt, die zwei Disketten in die Laufwerke gesteckt, und ab geht's! Nicht lange allerdings. Man glaubt gar nicht, wie schnell Mutter Erde einen wieder hat! Es hilft nichts – man muß die vier Kapitel des Handbuches durcharbeiten. Im ersten Kapitel lernt man Cockpit und Abheben mit der F-16 kennen. Ist man in der Luft – und bleibt auch dort – hat das zukünftige Fliegeras seine erste Begegnung mit einer feindlichen MIG. Der Pilot wird sie zwar noch nicht in Augenschein nehmen können, aber er sieht zumindest den Blip auf dem Radar.

Dieser Simulator verzichtet auf die Wahlmöglichkeit zwischen Flugplatz oder Flugzeugträger. Gestartet und gelandet wird nur auf einem Luftwaffenstützpunkt. Landungen werden durch die elektronische Landehilfe wesentlich erleichtert. Allerdings ist es wichtig, auch das manuelle Landen zu beherrschen — für den Fall, daß die Landehilfe in einem zukünftigen Luftkampf beschädigt wird.

Im zweiten Kapitel macht man sich mit dem Keyboard vertraut. Über 60 Tastenkommandos müssen intensiv geübt werden, damit man im Luftkampf oder beim Abschütteln von Boden-Luft-Raketen überhaupt eine Chance hat. Als eine der wichtigsten Tastenkombinationen erweist sich am Anfang Ctrl-E. Sie löst den Schleudersitz aus. Dann werden alle verfügbaren Waffen vorgestellt. Die Bewaffnung richtet sich nach der jeweiligen Mission. HUD (Head-Up Display) und Hauptkontrolleinrichtungen des Cockpits sind ausführlich erklärt. Die Anzeigen auf einen Blick zu erfassen, ist im kombinierten Luft-Luft- und Luft-Boden-Kampf unerläßlich. Da kann es vorkommen, daß man bis zu vierzig verschiedene Anzeigen beachten muß. Doch damit nicht genug! Auf der linken und rechten Seite des Cockpits sind weitere Anzeigen, die in Gefahrensituation zu kontrollieren sind. Blickrichtungen werden mit den Nummerntasten des Ziffernblocks bestimmt. Dabei kann man nicht nur den Rundblick aktivieren, sondern auch – wie beim „Jet“ – einen Track-, Tower- und Look-Down-View einschalten. Eine Besonderheit ist der Satellitenblick, der mit der Taste U aufgerufen wird.

Im dritten Kapitel werden die einzelnen militärischen Ränge, Auszeichnungen und Kampfeinsätze beschrieben. Es gibt sechs Ränge, vom Lieutenant bis zum Colonel. Je höher der gewählte Rang desto schwieriger ist das Fliegen. Braucht der Lieutenant sich kaum Sorgen um eventuelle Crash-Landings machen, hat der Colonel alle Hände voll zu tun, die Maschine zu fliegen und sich gleichzeitig seiner Gegner zu erwehren. Mit „F16-Falcon“ sind zwölf Missionen zu erfüllen. Sie reichen vom Milk Run-Training über Rattlesnake Roundup – nächtlichem Bombardement von Bodenzielen – bis zum Grand Slam. Im Grand Slam wird das fast Unmögliche verlangt; vier russische MIGs sollen abgewehrt werden.

Im letzten Kapitel werden überlebenswichtige Flugmanöver gelehrt: das High G Yo Yo (der Pilot verfolgt den Feind, schneidet eine Kurve ab und schießt von oben), die Immelmann-Kehre (gefährliches Ausweichmanöver mit Drehung und Looping) und die Scissors (Verfolgen und Ausweichen im Zickzack-Flug; der Feind wird von hinten angegriffen). Genauso intensiv muß man sich



Je höher der Rang und je schwieriger die Mission, desto sicherer der Crash.



Die Kollegen haben sich schon im Waffenarsenal bedient. Der Spieler muß mit altmodischen Waffen vorlieb nehmen.



Brücken sind strategisch wichtig. Dem F-16-Piloten sind sie willkommene Übungsziele, da sie nicht bewacht werden.

mit den Angriffstaktiken beim Luft-Boden-Kampf befassen. Von der Taktik hängt ab, ob und wie schnell man Bodenziele, also Gebäude, Brücken, Convoys, feindliche Stellungen (animiert) trifft. Die entsprechenden Flugmanöver lernt man mit einem Wingman, einem Begleitflugzeug.

Sobald der Pilot das Flight-Manual gründlich studiert hat, alle Keyboard-Commands und Flugmanöver beherrscht, kann er den ersten Flug mit Feindberührung wagen. Vor dem Start trägt er seinen Namen in der Rookie-Liste ein, wählt einen militärischen Rang, einen Auftrag und die Anzahl der MIGs, mit denen er es aufnehmen will. Maximal können drei MIGs angegeben werden, nur beim „Grand Slam" sind es vier. Dann muß das Flugzeug bewaffnet werden.

Im Waffenarsenal wählt man die Waffen unter Aufsicht eines kleinlichen Sergenanten aus. Dabei ist aber keineswegs sichergestellt, daß man die modernsten Waffen erhält. Piloten, die eher gestartet sind, haben sich schon bedient. So muß man mit dem auskommen, was noch übrig ist. Der Sergenant betrachtet jeden der stählernen Vögel als sein Privateigentum und wird bösartig, wenn man eines seiner Babys lädiert. Piloten, die es wagen, eine Falcon zu beschädigen, landen vor dem Kriegsgericht.

Wenn die Maschine beladen ist, werden mit der Maus (rechte Maustaste) und über Pull-Down-Menüs Optionen eingestellt. Es wird bestimmt, ob mit Maus, Joystick oder Tastatur gelenkt werden soll. Ferner lassen sich grafische Darstellung der Landschaft, Sound und Gefährlichkeit des Gegners festlegen.

Die F-16 rollt zur Startbahn. Nach einem kurzen System-Check gibt man vorsichtig Gas. Wenn die Schubkraft den Sollwert erreicht hat, werden die Bremsen gelöst. Ab geht's! Der Runway ist kurz. Bei schwerer Ladung muß der Nachbrenner gezündet werden. Mit einer Mindestgeschwindigkeit von 150 Knoten kann das Flugzeug abheben. Das Fahrwerk wird eingezogen und die Maschine behutsam auf Höhe gebracht. Mit den Tasten F3 und F4 kann die Lenkung empfindlicher eingestellt werden. Eine Empfindlichkeit vom Wert 9 (wird im Cockpit angezeigt) kommt der realen Reaktionsgeschwindigkeit einer Falcon am nächsten. Alles klar? Zielfindungs-, Angriffs-, Abwehr- und anderen Bordcomputersysteme unter Kontrolle? Auf zum großen „Rattlesnake Roundup" und Hals- und Beinbruch!

Für Amigabesitzer mit einem Ein-Megabyte-RAM bietet „F-16 Falcon" ein Extra: sie können die „Black Box" zur Aufzeichnung der eigenen und der gegnerischen Flugmanöver benutzen. Anhand dessen lassen sich Flugfehler erkennen und ausmerzen.

„F-16 – Falcon" ist einer der kompliziertesten und realistischsten Kampfflugsimulatoren. Der Hobby-Pilot vor dem Amiga bekommt eine leise Ahnung von der Komplexität einer Luftkampfausbildung. Und es erscheint nicht mehr verwunderlich, daß Jagdflieger schon mit 45 Jahren in den Ruhestand geschickt werden.

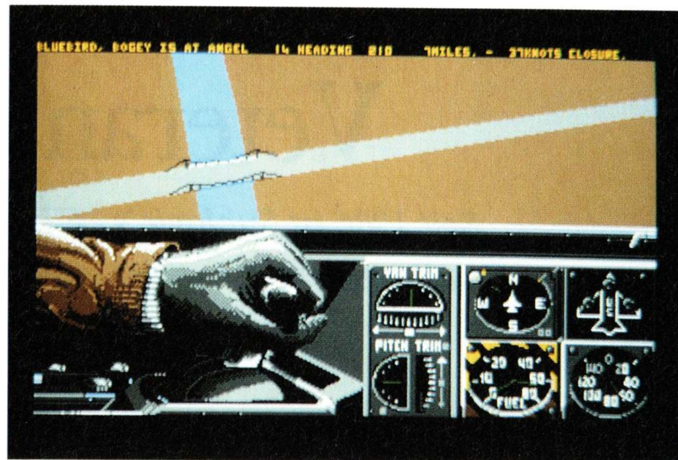
Grafik und Sound sind bei diesem Simulator gut gelungen. Um die Rechengeschwindigkeit zu erhöhen, wird die einfachste Landschaftsdarstellung eingeschaltet.

„F-16 Falcon" kann man auch zu zweit spielen. Dazu wird über Modem ein weiterer Amiga, ein Atari ST oder ein Macintosh angeschlossen.

(R. de Flers)

Preis: 100 Mark
Hersteller: Microprose
Vertrieb: Leisuresoft

Spaß	9
Grafik	8
Sound	7
Idee	6
Schwierigkeit	10



Ein Blick zur Seite zeigt weitere Meß- und Kontrollinstrumente.



Eine russische MIG — erkennbar an der roten Flugzeugnase — zieht haarscharf an der F-16 vorbei.



Treffer! Eine russische MIG explodiert. Doch ist dies kein Grund, leichtsinnig zu werden. Schon im nächsten Moment kann die nächste MIG das Feuer auf die F-16 eröffnen.

Flightsimulator II

Veteran der Lüfte

Seit 1986 ist Sublogic's „Flightsimulator II“ auf dem Markt. Das Programm ist weder schnell noch bunt, aber sehr realistisch und immer noch aktuell.

Heftige Turbulenzen schütteln die Cessna. Der Pilot bleibt ruhig. Durch geschicktes Gegensteuern gelingt es ihm, die Maschine unter Kontrolle zu bringen. Er läßt das Flugzeug 2000 Fuß absinken, um die unruhige Windschicht zu verlassen. Hinter Nebelschleiern ist der Tower des Merrill C. Meigs Airport zu erkennen. Die Cessna schwenkt um 40 Grad, fliegt einen leichten Bogen und nimmt Kurs auf Chicagos größten Flughafen, O'Hare International. Bevor der Pilot in den Landeanflug übergeht, läßt er sich die lokalen Wetterverhältnisse durchgeben. Dazu wird das COM-Funkgerät mit einem Klick auf das COM-Feld auf die entsprechende Frequenz eingestellt. Auf dem Bildschirm erscheint der Bericht: „O'Hare International Airport Information Bravo 17:00 Zulu Weather — Measured Ceiling 1400 Overcast — Visibility 4 — Temperature 7 — Wind 210 at 400 — Altimeter 29.25 — Landing and Departing Runway 22 — Advise Controler on initial Contact you have Bravo....“. Dichter Bodennebel und eine niedrige Wolkendecke über dem Runway machen eine Sichtlandung unmöglich. Der Pilot muß seine Maschine im Blindflug zu Boden bringen. Die beiden Navigationsgeräte NAV1 und 2 werden mit Mausklicks auf den richtigen Leitstrahl des UKW-Drehfunkfeuers (Feld VOR) gebracht. Sinkgeschwindigkeit o.k.! Es sind keine Probleme bei der Landung zu erwarten. Konzentriert beobachtet der Pilot die Anzeige des DME-Entfernungsmeßgerätes, um den Abstand zum Funkfeuer zu ermitteln. Bis zum Runway sind nur noch wenige Meilen zu fliegen. Die Flughöhe beträgt knapp 1500 Fuß. Im Landeanflug werden die Landeklappen mit Anklicken von FLAPS auf 20 Grad Neigung gebracht. Mit einem Druck auf die Taste U wird das Fahrwerk ausgefahren und mit der Maus im Feld Throttle (TH) langsam auf 60% Schubkraft reduziert. Die Cessna nähert sich dem Funkfeuer und verliert Höhe. Plötzlich ist die Landebahn zwischen den



Im Beobachtermodus sieht der Pilot seinen Learjet. Die Entfernung ist variabel.



Bei schönem Wetter sind Landschaftsdetails wie die Festung Alcatraz, Golden Gate und die Hochhäuser San Franciscos klar erkennbar.

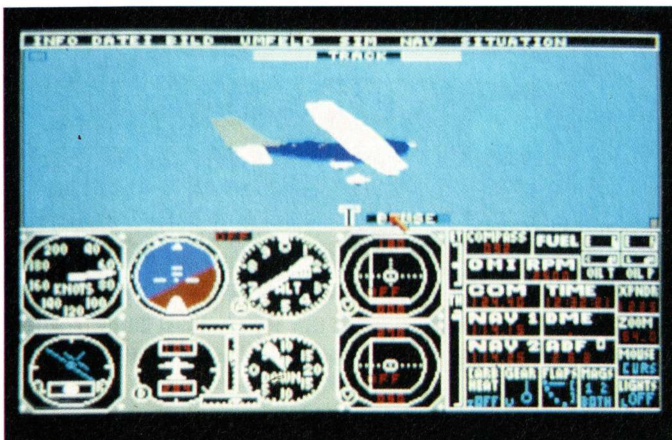


Instrumenten-Check vor dem start. Ist der Tank voll? Stimmen Öldruck und -temperatur?.

Nebelschwaden erkennbar. Schnell reduziert der Pilot den Schub auf 40% und stellt die Landeklappen auf 40 Grad Neigung. Die Cessna sinkt jetzt sehr schnell. Kurz vor dem Aufsetzen wird die Flugzeugnase ein wenig hoch gezogen; dazu zieht man den Joystick nach hinten. Geschafft! Der Pilot nimmt den Schub weg und bremst mit der Taste 1 des Zehnerblocks. Über die Rollbahnen geht es zum Hangar. Dort wird die Maschine aufgetankt, durchgecheckt und gewartet; dazu begibt man sich zu den mit F markierten, rechteckigen Arealen des Airports. Erleichtert läßt sich der Pilot vor dem Amiga in seinen Sessel zurückfallen und speichert die Position auf Diskette ab. Sublogic's „Flightsimulator II" ist seit 1986 auf dem Markt. Die meiste Software dieses Jahrgangs ist veraltet und schon lange vergessen. Der Flight II ist immer noch beliebt. Mit sechs Bildern in der Sekunde gehört das Programm zwar nicht zu den schnellsten, doch werden beim Flight II mehr Umweltbedingungen als bei anderen Flugprogrammen simuliert. Mit Hilfe übersichtlicher Menüs lassen sich Wolkenfelder und Bodennebel, Turbulenzen und Windstille, ja sogar Jahreszeiten simulieren. Man darf sich also nicht wundern, wenn während des programmierten Winters die Pisten vereist sind oder der Motor nicht anspringt. Damit ist das Potential jedoch nicht erschöpft. Der Flight II simuliert auch Unerfreuliches wie Motorschäden, Instrumentenausfälle und Wetterstürze. Professionell wird Flight II durch realistisch arbeitende Navigationssysteme, die dem Anpeilen von Flughäfen dienen. Bei Bedarf lassen sich die Navigationssysteme mit dem Autopiloten koppeln; dem gestreßten Piloten wird Arbeit beim Navigieren abgenommen. Mit Flight II läßt sich die Cessna 182 oder der Learjet 25G simulieren. Die Option „Jagdflieger" bietet Kriegerisches. Bei dem Jagdflieger handelt es sich um einen Doppeldecker aus dem Ersten Weltkrieg. Damit fliegt man über Feindgebiet und schießt auf Fabriken, Treibstofflager und Fliegerhorste des Gegners. Mit dem Flight II läßt sich ein Gebiet von einer Viertel Milliarde Quadratkilometern überfliegen! Die Programmdiskette bietet über 200 Flughäfen. Zusätzlich kann man Landschaftsdisketten kaufen. Für die Amigaversion gibt es Scenary Disks von Japan, Westeuropa und mehreren US-Staaten. Flight II ist komplett ins Deutsche übersetzt worden. Die beiliegenden Flugkarten sind unerläßlich zur Information über Lage und Charakteristika der Flughäfen.

Info:
technicSupport, Bundesallee 36–37, 1 Berlin 31, Tel.: 030/862 13 14, Preis: 290 Mark incl. dt. Anleitung

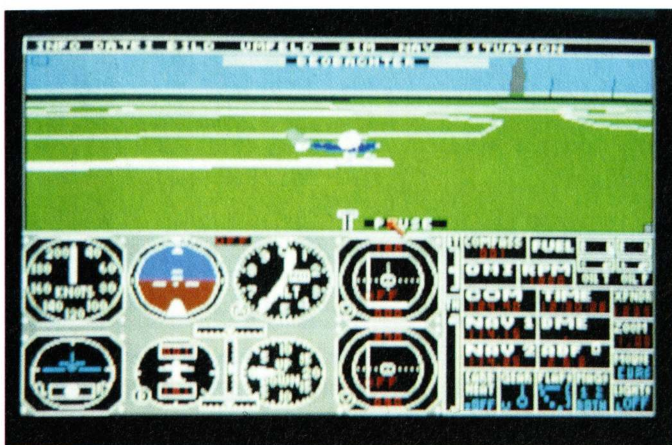
(Andreas Fuchs)



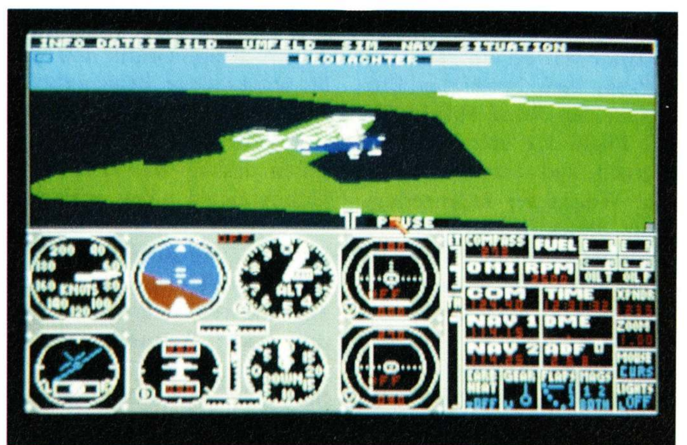
Selbst mit Gegenverkehr ist zu rechnen. Hier fliegt eine Cessna des Typs Turbo Skylane RG II vorbei.



Landschaften sind vereinfacht, aber in realistischen Größenverhältnissen dargestellt.



Grob dargestellte Merkmale helfen bei der Orientierung. Hier erkennt man Chicagos Sears Tower am Horizont.



Flugtechnische Daten werden am Instrumentenbrett angezeigt. Die momentane Bildeinstellung ist in der Zeile unterhalb des Menü-Balkens erkennbar.

Flightsimulator III

Mit Flight III zu neuen Höhen

Unter den Flugsimulatoren ist Flight II der professionellste — jedoch nicht mehr lange. Die PC-Version von Sublogic's Flight III zeigt, daß die Fliegerei komplexer und vielseitiger wird.



Vor dem Start auf Chicagos Airport Meigs: Maschinen- und Instrumentencheck.

Mit drei Disketten, erweitertem Kartenmaterial und einem gebundenen Handbuch präsentiert sich die deutsche PC-Version des Microsoft Flight Simulator III. Wer nicht im Besitz eines Bridgeboards ist, bekommt zumindest einen Vorgeschmack auf die in Arbeit befindliche Amiga-Version.
Microsoft Flight II ist zum Klassiker geworden. Mit seiner Vielfalt sprengt das Programm den Rahmen handelsüblicher Spiele. Dem Hobbypiloten bietet die Simulation sowohl seriöse Unterhaltung als auch viele kurzweilige Stunden vor dem Computer. Der ökonomische Erfolg rief die Konkurrenz auf das Schlachtfeld der freien Marktwirtschaft. So mancher Software war der Ehrgeiz anzumerken, dem bewährten Programm den Rang abzulaufen. Mit Flight III hält Microsoft dagegen und erzeugt prompt neue Wogen der Begeisterung. Trotz dieses berechtigten Enthusiasmus stellt sich die Frage, ob die Unterschiede gegenüber der Vorversion so gravierend sind, daß sich der Umstieg lohnt. Eines ist sicher — geändert hat sich einiges. So ermöglicht der Verzicht auf einen Kopierschutz die problemlose Installation auf Festplatte. Die Anlage eines Verzeichnisses und das Übertragen sämtlicher Dateien in das neue Verzeichnis genügen.
Beim Aufruf des Programms meldet sich ein Installationsmenü, das die Unterstützung verschiedenster Grafikformate ermöglicht. Von Monochrome bis VGA werden alle gängigen Modi unterstützt und genutzt. Gleichzeitig kündigt sich an, daß penible Hobbypiloten die Möglichkeit haben, ein Bordtagebuch zu führen.

Option Sprühflieger: Im Tiefflug Schädlinge vernichten

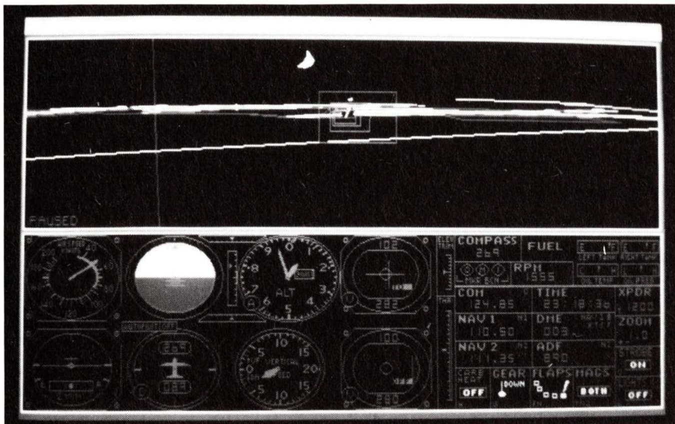
Maus und deutsche Tastaturversion werden unterstützt.
Ansprechende Grafik und völlig umgestaltete Benutzeroberfläche beweisen, daß nicht nur Kosmetik betrieben wurde, und lassen deshalb das Herz jedes Hobbypiloten höher schlagen. Vorbei sind die Zeiten des mittels Esc-Taste aufrufbaren Editors — eine Menüleiste mit fünf anwählbaren Untermenüs (Modus, Sicht, Umwelt, Simulation, Navigation/Kommunikation) gibt ein deutliches Zeugnis anwenderbezogener Software-Ergonomie. Hinter unscheinbar wirkenden Menüpunkten verbirgt sich eine Detail-Liebe, wie man sie selten findet.



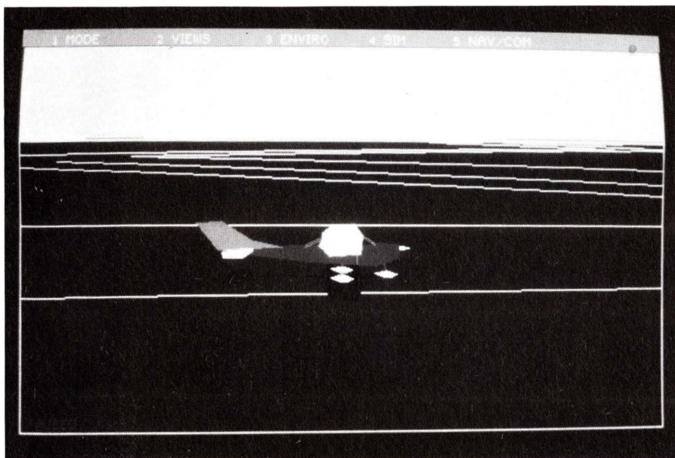
Kurz nach dem Abheben wirft der Pilot einen Blick zurück auf die Runway. Dazu wird das Heckfenster aktiviert.



Mit Navigations- und Landehilfen ist ein direkter Landeanflug möglich. Der Flughafen ist noch nicht erkennbar; ein Leitstrahl weist den Weg.



Landeanflug kurz vor Mitternacht. Funkfeuersignale leiten das Flugzeug sicher zu Airport und Landebahn.



Die Cessna kurz vor dem Aufsetzen. Mit einem Druck auf die Taste G wird das Fahrwerk ausgefahren.

Ein Blick ins Modus-Menü läßt die Features ahnen, die Version III prägen. Drei simulierte Flugzeuge (Cessna, Lear-Jet, Doppeldecker), verschiedenste Spielarten (Weltkrieg-1-Luftkampf, Geschicklichkeitsflug, Multiplayermodus, Sprühfliegen) und die Möglichkeit, eigene Demos aufzuzeichnen sind nur ein Teil davon.
Landung auf einem Flugzeugträger gefällig? — Bitte sehr, die Nimitz steht zur Verfügung. Oder wie wär's mit einem Flug unter der Golden-Gate-Bridge hindurch? — Auch das ist mit entsprechendem fliegerischen Geschick machbar.
Da darf dann auch neueste Technologie nicht fehlen. Electronic Flight Instrument System (EFIS) und Command Flight Path Display (CFPD) heißen die Zauberworte. Damit hat der Computer auch ins Cockpit Einzug gehalten. Mittels einer Projektion auf die Cockpit-Sichtscheibe können auf diese Weise Flugkurse und Leitstrahlen wie zum Beispiel beim Instrumenten-Landesystem sichtbar gemacht werden. So erscheint bei einem ILS-Landeflug der Gleitstrahl als Korridor, dem das Flugzeug nur noch folgen muß, um auf die Landebahn zu kommen. Aufgrund von Computerberechnungen wird unter Verwendung der entsprechenden Frequenz des ILS-Systems der Flugweg auf einen durchsichtigen Schirm der Cockpitwindschutzscheibe (Head-Up-Display = HUD) projiziert und dient als verbesserte Orientierungshilfe.
Neu ist auch der Fluglehrermodus, in dem vom Rollen auf dem Flugfeld bis zum Landeanflug alles unter Anleitung geübt werden kann. Dabei wird zunächst gezeigt, wie's geht, und dann wird geübt.
Moderne Fenstertechnik kennzeichnet das zweite Menü (Sicht). Bis zu drei Fenster zum Beispiel für Front-, Rück- und Kartensicht können definiert, größenmäßig variiert und beliebig verschoben werden. Wer gerne vollständige Sicht hat, schaltet die Instrumentenanzeige ab und genießt vollen Ausblick auf die Landschaft.
Towersicht, Tracking und Ansicht aus der Perspektive eines Begleitflugzeugs ergänzen die vielfältigen Möglichkeiten der Betrachtung.
Liebe zum Detail kennzeichnet auch das Umwelt-Menü. Bereits bekannten Einstellungen für Wolken-, Wind- und Turbulenzschichten wurde Beachtenswertes zugefügt. Die Wolkendichte von „klar" über 1/8, 2/8,... kann bis hin zu „bedeckt" beliebig gewählt werden. Bei Teilbedeckungen erscheinen Wolken als weiße Bällchen.
Nachtflieger können bei klarem Himmel Mond und Sterne bewundern — die Gestirne wurden bis zur dritten Größenordnung originalgetreu nachgebildet.

Die Bahnen von Sonne, Mond und Sternen werden simuliert

In Menüpunkt 4 werden die Simulationparameter geregelt. Erstmals läßt der Flugsimulator Empfindlichkeitseinstellungen bezüglich der Reaktionen von Tastatur und Joysticks zu.
Apropos Eingabegeräte: Die Maus wurde mit einer Doppelfunktion versehen; die linke Taste gilt nach wie vor der Bedienung der Instrumente, während die zweite Taste die Maus zum Steuern des Flugzeugs befähigt.
Crash-Piloten können aufatmen: Der Flugsimulator bietet die Möglichkeit, Abstürze zu ignorieren, ohne den früher üblichen Reset und Rückkehr zur Starteinstellung durchführen zu müssen. Wer's krachen läßt, kann sich seine Absturzkurve als Diagramm ausgeben lassen — lediglich Crash-Detect und Analyse müssen eingeschaltet sein. Das letzte Untermenü dient der Navigation und Kommunikation. Neben den aus der Vorversion bekannten Flugräumen steht dem Anwender zusätzlich die San-Francisco-Area zur Verfügung. Besitzer der deutschen Version können in Mitteleuropa und Süddeutschland fliegen — der Münchner Olympiaturm läßt grüßen...
Wer bei den vielen Neuerungen Angst um die gewohnte Bedienung bekommen hat, sei beruhigt: Alle Cockpit-Kontrollen sind mit identischen Befehlen wie bei der Vorversion ansteuerbar — das Umlernen beschränkt sich auf jene Neuerungen, die hauptsächlich im Bereich der Menüsteuerung liegen. — Die Mühe lohnt sich in jedem Fall.
Wer über die Szenerie-Disketten verfügt, kann sowohl diese als auch alte, selbstgestrickte Flugmodi mit der neuen Version weiter benutzen — lediglich die Szenerie-Disketten müssen mit einer mitgelieferten Konvertierungsroutine bearbeitet werden.
Die Anzahl der Flughäfen und der Funkfeuer wurde erhöht. Chicago OHare ist inzwischen mit einem ILS-System (Frequenz 110.5) ausgerüstet. In diesem Zusammenhang verbergen sich viele nützliche Informationen in der Readme-Datei auf einer der Disketten.
Ein Umstieg auf die neue Version ist also empfehlenswert. Für 149 Mark erhält man ein uneingeschränktes Flugvergnügen.

Mit einem **Maxx-Steuerknüppel** (siehe Titelbild) fühlt man sich wie im Cockpit. Der Maxx-Flugsimulator-Joystick kann an den Amiga 2000 (mit entsprechender Joystick-Karte) und den PC angeschlossen werden. Erhältlich bei Peksoft für 350 Mark.

(Dominic Hammon)

Wir sind ein Mitglied der International Data Group, der Welt größter Verleger für computerbezogene Informationen. Die Gruppe veröffentlicht 90 Computer-Publikationen in 33 Ländern. 14 Millionen Menschen lesen eine oder mehrere Publikationen dieser Gruppe pro Monat. Die Mitglieder sind am CW-Communications International News Service angeschlossen, einem täglichen Nachrichtendienst für die aktuellsten Meldungen aus dem internationalen DV-Geschehen.

Wir suchen für

MS-DOS WELT

AMIGA WELT

Programmautoren, Testexperten, freie Mitarbeiter, Bastler.

Sie sollten Erfahrung im Umgang mit Computern, aber auch Kenntnisse der zugehörigen Hard- und Software haben. Kontaktfreude und Grundkenntnisse der gängigsten Programme wie Textverarbeitung und Grafik setzen wir voraus. Die Honorierung entspricht der vergleichbarer Verlage. Eventuell anfallende Reisekosten werden ersetzt. Eine Textprobe von einem möglicherweise bereits veröffentlichten Beitrag würde uns natürlich sehr freuen. Für einen ersten Kontakt steht Ihnen zur Verfügung:

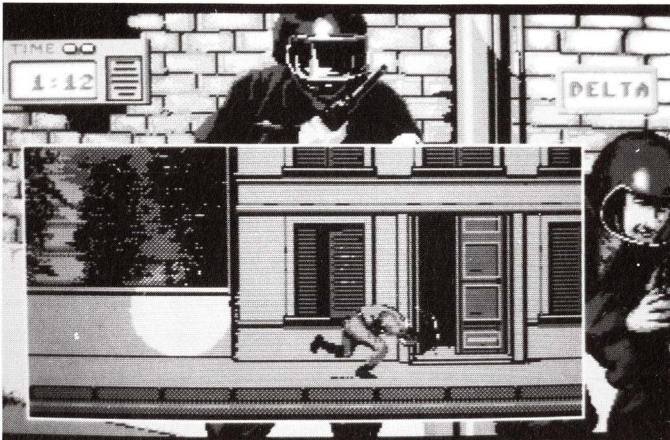
Siggi Pöschel
Tel. 0 89/3 60 86-210
CW-Publikationen Verlagsgesellschaft mbH
Rheinstr. 28
8000 München 40

Hostages

Sturm auf die Botschaft

Terroristen haben eine Botschaft überfallen und Diplomaten als Geiseln genommen. Jetzt verbarrikadieren sich die Gangster im Botschaftsgebäude und erpressen die Regierung. Die Staatshäupter wollen beweisen, daß sie nicht erpreßbar sind und beauftragen eine Spezialeinheit mit der Stürmung der Botschaft. Der Spieler als Leiter des Kommandos soll nun drei Elitekämpfer in die Botschaft bringen und mit deren Hilfe alle Geiseln befreien. In der ersten Spielsequenz wird ein Scharfschütze nach dem anderen über die Straße zur Botschaft geführt. Wenn die drei Männer ihre Plätze eingenommen haben, beginnt Spielstufe zwei. Über der Botschaft kreist ein Hubschrauber. Man sieht drei Männer, die Elitekämpfer, herausspringen. Sie werden auf dem Dach verteilt. Mit den Tasten F1 bis F6 läßt sich zwischen den drei Scharfschützen hin- und herschalten. Die Schützen müssen nun die Fenster der Botschaft zerschießen, um Einlaß zu schaffen. Mit Seilen klettern die Kämpfer vom Dach herab und in die Fenster hinein. Das Szenario wechselt. Die Geiseln müssen gefunden und in ein bestimmtes Zimmer in Sicherheit gebracht werden. Begegnet man einem Terroristen, so zielt man mit dem Fadenkreuz und feuert mit dem Maschinengewehr. Mit einem Schrei sackt der Getroffene zusammen und liegt zusammengekrümmt am Boden — eine Szene, die um einige Grade zu makaber geraten ist. Hier wird vielmehr Wert darauf gelegt, daß alle Geiseln in einem Zimmer gesammelt und vor den Terroristen geschützt werden. Am Ende des Spiels ist nämlich in der Zeitung eine Kritik zur Geiselbefreiung zu lesen. Eine gute Kritik gebührt dem Spiel selbst. Grafik, Musik und Soundeffekte verdienen ein Lob. Konzeptionell wurde hier einmal eine ungewöhnliche Schiene gefahren: Nicht nur am Joystick soll man flink sein, sondern man soll auch strategisches Gespür zeigen und seine Mission ohne Blutvergießen erfüllen.

(C. Borgmeier)



Scharfschützen geben Elitekämpfern Feuerschutz.

Preis: 80 Mark
Hersteller: Infogrames
Vertrieb: Bomico

Spaß	8
Grafik	9
Sound	9
Idee	8
Schwierigkeit	7

Ingrid's back

Gnomhausen wird saniert

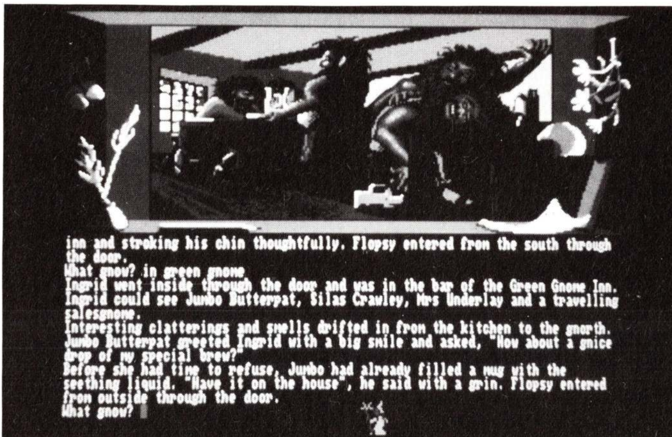
Schön scheußlich! Das ist wohl der erste Eindruck, den Spieler von der Gnomdame Ingrid haben. Ingrids Familie findet das auch. Im Adventure „Gnome Ranger" konnte man miterleben, wie die Gnome sich eine Erholungspause von Ingrid verschaffen. Mit einem Teleport-Gerät schickten sie die Unausstehliche für drei Monate in die Wildnis. Die Zeit ist um. Ingrid ist wieder da. Aber diesmal werden die Gnome ihre Ingrid als sozial engagierte Kämpferin erleben. Sie können sich freuen, denn „Ingrid's back"!

Der Spieler alias Ingrid steht vor folgendem Problem: Little Moaning, die verträumte Gnomstadt, ist auf illegale Art und Weise in die Hände des geldgierigen Jasper Quickbuck gefallen. Der romantische Ort soll an Grundstücksmakler verhökert, saniert und in ein Yuppie-Viertel verwandelt werden.

Ingrid tut nun das, was auch Bürgerinitiativen als erstes unternehmen; sie sammelt Unterschriften. Im Spiel schickt man Ingrid von Haus zu Haus und tippt immer dasselbe Sprüchlein ein — „Offer a petition to . . .". Verwandte und Bekannte unterschreiben prompt. Nur ein eigensinniger Kauz will nicht auf der Liste unterzeichnen. Allerdings weiß er die Namen all derjenigen, die noch nicht befragt wurden. Und mit ein wenig rhetorischem Geschick entlockt Ingrid ihm die Namen der betreffenden Bürger. Mit dem Befehl „find . . ." werden die Leute aufgespürt. Sobald die Unterschriftenliste komplett ist, wird sie der „great aunt" übergeben. Im zweiten Teil des Adventures verteidigt Ingrid die elterliche „Gnettlefield Farm" gegen Grundstücksspekulanten. Wenn sie aus diesem Kampf siegreich hervorgeht, darf sie endlich ihren rauhen Charme einsetzen und den Fiesling Jasper Quickbuck um Ruhe und Verstand bringen. Es wäre ja gelacht, wenn Ingrid ihn nicht kleinkriegte!

„Ingrid's back" ist ein Adventure wider den tierischen Ernst. Humorvolle Comic-Zeichnungen und die eigentümliche Gnomspache - in Englisch - machen das Spiel zum witzigen Zeitvertreib. Auf gut gnomisch: Wer sich „gnäher" mit Ingrid und ihrem „gniedlichen" Hund Flopsy befassen will, sollte „gnicht" länger warten.

(mms)



Im Gnome-Inn geht es rauh, aber herzlich zu.

Preis: 70 Mark
Hersteller: Level 9 Computing
Vertrieb: Leisuresoft

Spaß	8
Grafik	6
Parser	8
Idee	8
Schwierigkeit	7

MS-DOS WELT

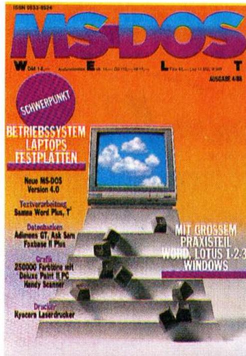


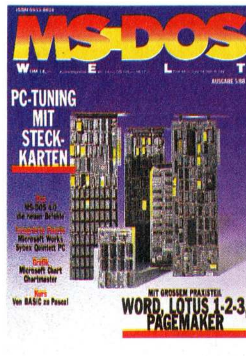


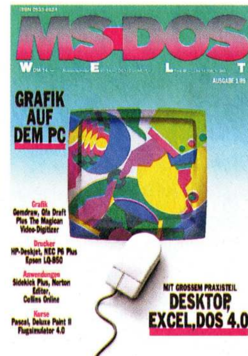




















MS-DOS WELT

Die Zeitschrift für die Anwender des MS-DOS-Betriebssystems bietet ausführliche Informationen für den ambitionierten Privatnutzer und vermittelt umfassende, fundierte Kenntnisse für den täglichen Umgang mit dem Computer.

CW PUBLIKATIONEN

Bestell-Coupon

JA, ich bestelle aus Ihrem Angebot folgende(n) Titel:

____ Exemplar(e) MS-DOS WELT 1/88

____ Exemplar(e) MS-DOS WELT 3/88

____ Exemplar(e) MS-DOS WELT 4/88

____ Exemplar(e) MS-DOS WELT 5/88

____ Exemplar(e) MS-DOS WELT 1/89

____ Exemplar(e) MS-DOS WELT 2/89

zum Einzelpreis von ***DM 14,—.***

Gewünschte Zahlungsweise bitte ankreuzen:

☐ Per Verrechnungsscheck (liegt bei*) (Heftpreis(e) + DM 2,— Porto**)

☐ Per Nachnahme (Heftpreis(e) + DM 2,— Porto** + DM 1,70 Nachnahmegebühr)

*Auslandsbestellungen nur gegen Verrechnungsscheck

**Ab 2 Exemplaren erhöht sich der Portoanteil auf DM 3,—

Datum/Unterschrift

Meine Anschrift

Firma (falls Lieferanschrift)

Name/Vorname

Straße, Hausnr./Postfach

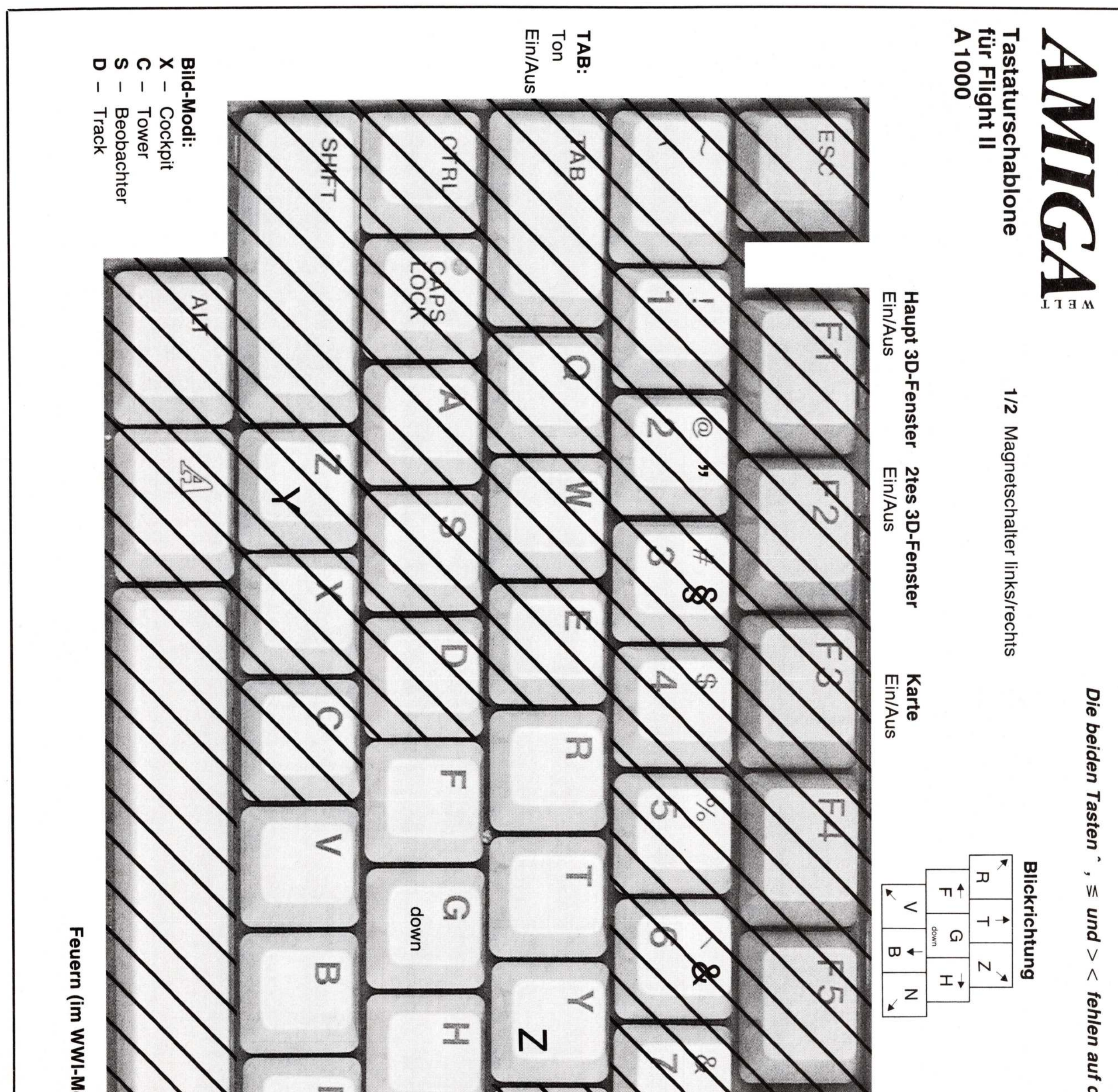
PLZ Ort

Branche/Funktion

Cockpit für Keyboard-Flieger

Damit die Orientierung nicht schon bei der Suche nach der richtigen Taste verlorengeht, sind hier Schablonen für den simulierten Blindflug (sprich ohne ständigen Sichtkontakt mit dem Handbuch) abgedruckt. Schablone ausschneiden, Tastenfeld herausnehmen und die zwei Teile zusammenfügen. Die Schablone läßt sich noch mit Pappe verstärken und mit Folie oder Klarlack überziehen. Für die 1000er-Besitzer mit amerikanischem Keyboard sind die deutschen Tastenbezeichnungen neben den englischen eingetragen. Mit Klarlack kann man die Blickfeldtasten dauerhaft schützen, ausschneiden und auf die Tastatur aufkleben.

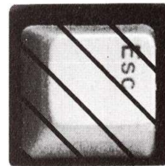
Teil 2 auf Seite 119



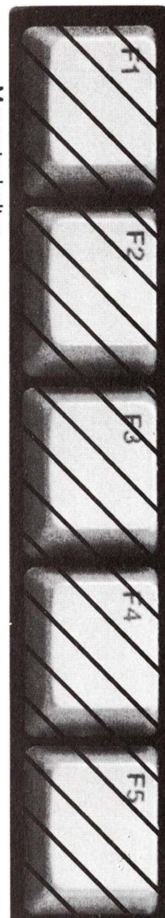
Teil 2 auf Seite 120

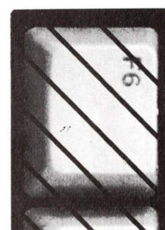
AMIGA WELT

Tastaturschablone für FLIGHT II A500/2000



Haupt 3D-Fenster Ein/Aus **2tes 3D-Fenster** Ein/Aus **Karte** Ein/Aus

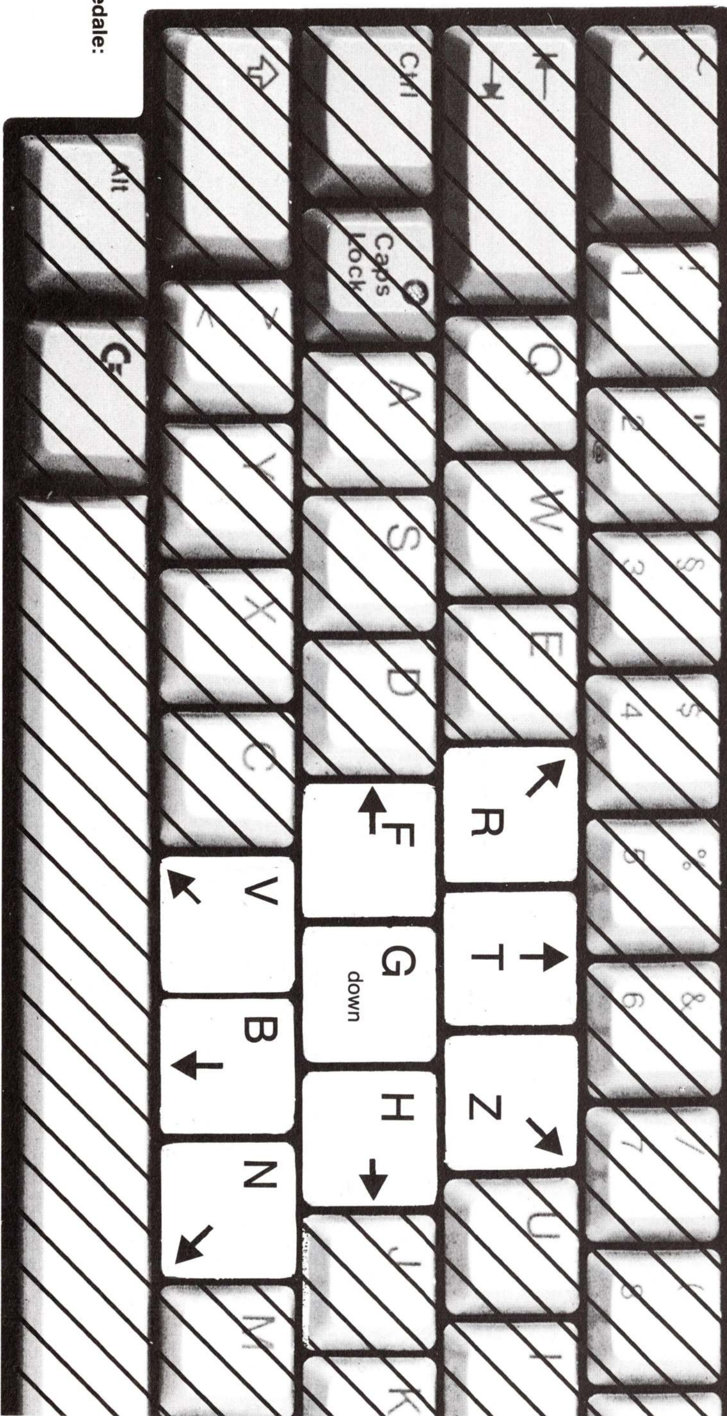




Magnetschalter
Links Rechts

TAB:
Ton Ein/Aus

Seitenruderpedale:
< – Links
> – Rechts



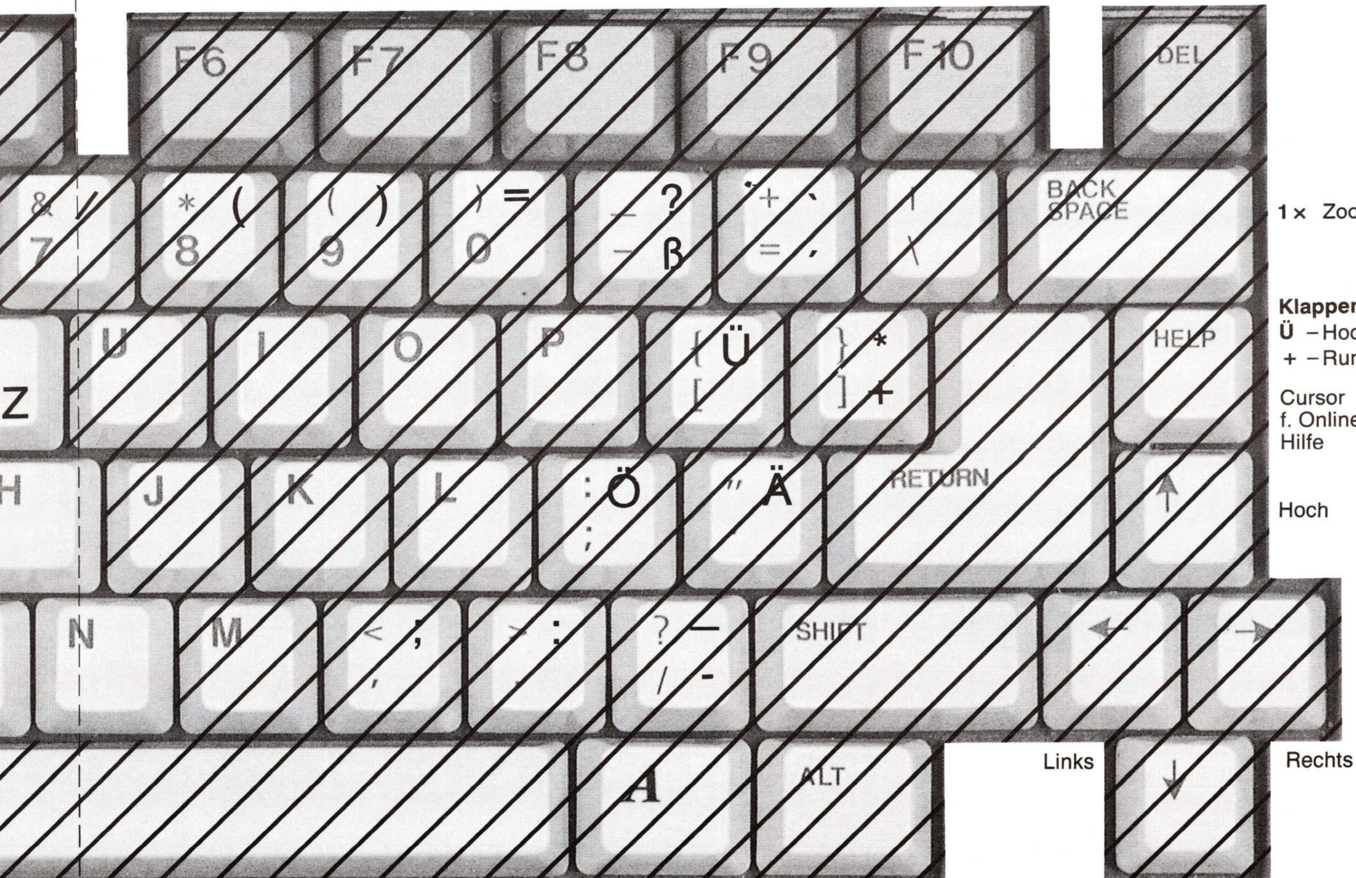
Feuern (im WWI-Modus)

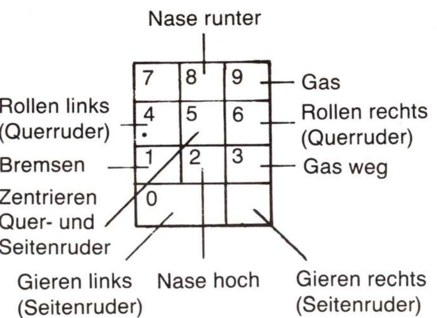
en auf dem 1000er Keyboard unter deutscher Belegung. Sie sind über die Kombination SHIFT,ALT und dem amerikanischen Äquivalent zu erreichen.

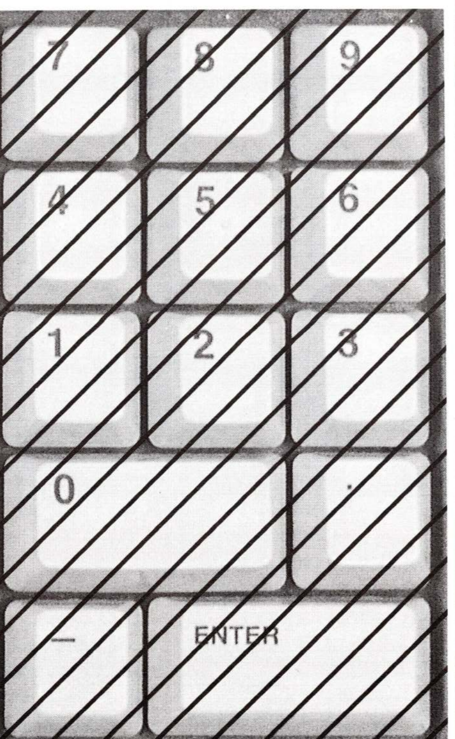
Nur im WWI-Modus:
Shift-W – Spielbeginn
Shift-E – Ergebnisbericht
Shift-X – Bombe
Leertaste – Feuern

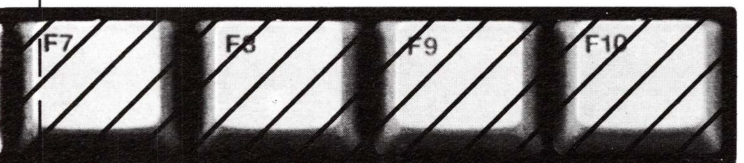
Sonstiges:
Z – Autopilot Ein/Aus
A – Laden der Flugsituation
Q – Speichern der Flugsituation
O – Schattierung Ein/Aus
L – Lichter An/Aus

I – Vergaser vorwärmen Ein/Aus
U – Fahrwerk Ein-/Ausfahren
9/0 – Trimmung hoch/runter
ß/` – Zoom fein fern/nah

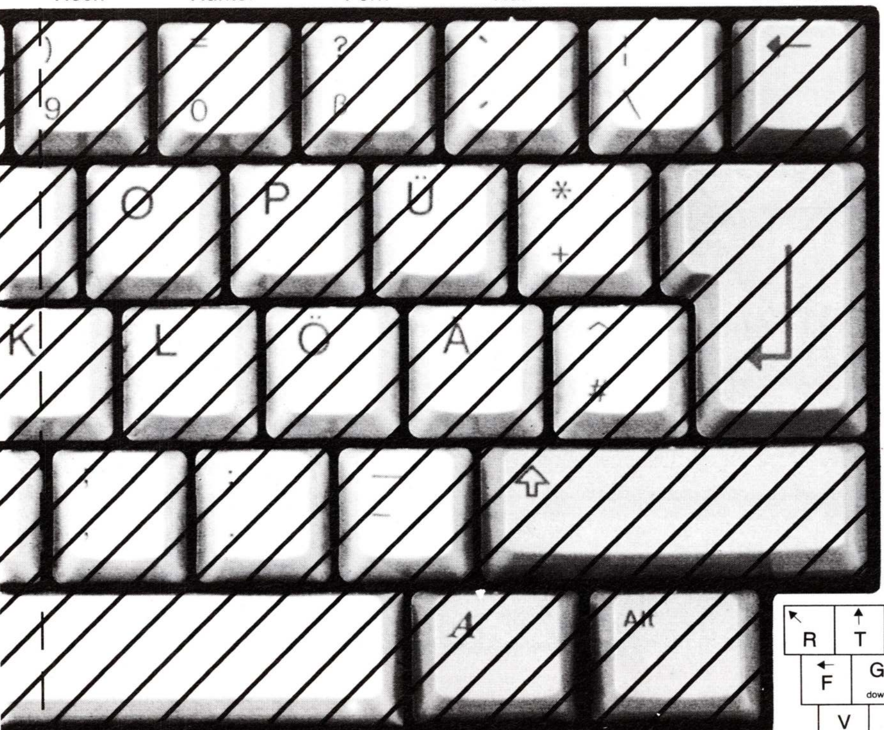








Klappen:
Ü – Hoch
+ – Runter



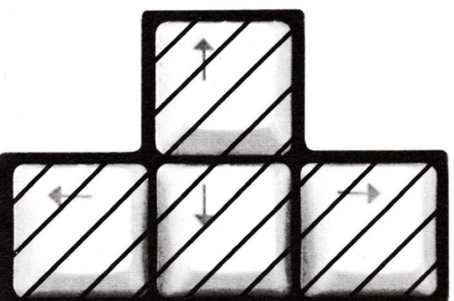


Blickrichtung

Nur im WWI-Modus:
Shift-W – Spielbeginn
Shift-E – Ergebnisbericht
Shift-X – Bombe
Leertaste – Feuern

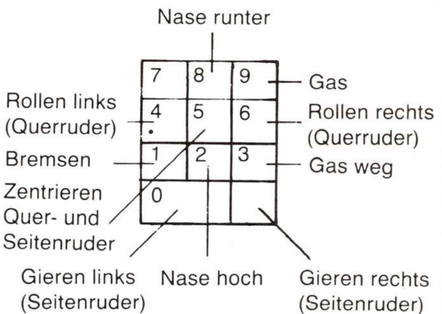
Bild-Modi:
X – Cockpit
C – Tower
S – Beobachter
D – Track

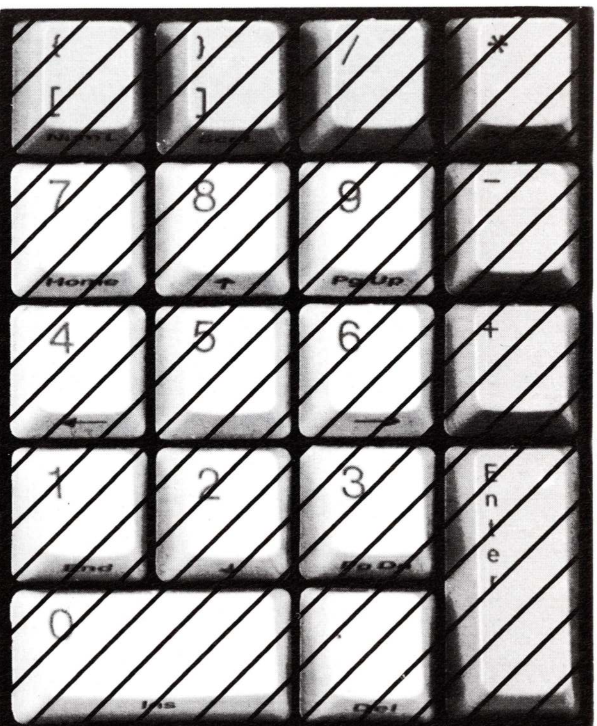




Blick-Feineinstellung

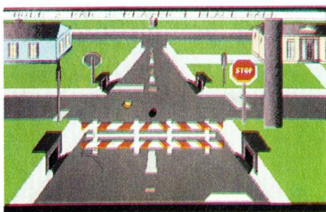
Sonstiges:
Z – Autopilot Ein/Aus
A – Laden der Flugsituation
Q – Speichern der Flugsituation
O – Schattierung Ein/Aus
L – Lichter An/Aus
I – Vergaser vorwärmen Ein/Aus
U – Fahrwerk Ein-/Ausfahren





Kurz notiert:

Magic Bytes setzte die epidemische Massensportart Minigolf wohnzimmergerecht für den Amiga um. Das Spiel erfordert enorme Kondition im rechten Zeigefinger sowie extreme Konzentration bei den kniffligen Bahnproblemen. **„Minigolf“** ist das ideale Spiel für Leute, die den Umgang mit der Maus intensiv trainieren möchten.



Das erfolgreiche Atari-ST-Spiel **„Netherworld“** von Hewson liegt endlich als Amiga-Konvertierung vor. In unglaublichem Tempo fliegt ein rotierendes



Raumschiff durch zahllose Labyrinthe. Auf der Suche nach Edelsteinen und Bonus-Icons muß der kleine Flitzer unterschiedliche steinerne Hindernisse überwinden.



Electronic Arts versoftete die Science-Fiction-Novelle **„Neuromancer“** von William Gibson. Der Spieler wird in das als trostlos gekennzeichnete Japan des 21. Jahrhunderts versetzt. Die Welt steht unter dem Kommando eines gigantischen Com-



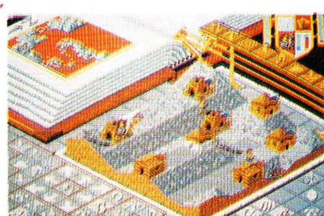
putersystems. Pfiffige Hacker - der Spieler wird zur Identifikation eingeladen - versuchen, in das elektronische Netz einzudringen. Wer über beste Software, Hardware und Informationsquelle verfügt, wird den Großrechner und damit die Welt regieren. Der Spieler „Neuromancer“ kostet 79 Mark und kommt von Rushware.



In **„Populous“** wird der Spieler zum Gott. Als Herrscher über die Welt lenkt er die Entwicklung zweier Völker — ein Volk



soll von Natur aus gut sein, das andere schlecht. Dennoch sind beide religiös, fanatisch und kriegerisch. Der zur Hybris verleitete Amiga-Freund muß eines



der Völker bevorzugen. Er sorgt dafür, daß es neues Land erobert, urbar macht und Technologien entwickelt, bis es imstande ist, das andere Volk zu vernichten. Die Aussicht auf derlei irdische Entwicklungen genießt man in 3D. Entgegen westlich monotheistischer Vorstellung kann „Populous“ von zwei Göttern gespielt werden, falls der zweite Gott einen Atari ST oder einen Amiga hat. Für 79 Mark bei Rushware zu beziehen.



Zauberhaftes kommt von Rainbow Arts: **„Spherical“**. Als Zauberer erforscht man die verschiedenen Ebenen einer unbekannten Welt. Dabei hilft eine magische Wunderkugel. Um zusätzlich übermenschliche Kräfte und Macht zu erlangen, sammelt der Zauberer Geheimschriften, Medaillons, Zauberstäbe und -tränke. Dadurch ist er bald in der Lage, mit Feuerbällen zu schießen, Steine herzustellen und zu teleportieren. Steine sind die Schlüssel zu den Ebenen. Für 60 Mark bei Rushware.



Cinemaware entführt den Spieler mit **„Lords of the Rising Sun“** ins Japan des 12. Jahrhunderts. Die Story orientiert sich an historischen Ereignis-



sen. Es herrscht Bürgerkrieg. Der Spieler setzt alles daran Shogun zu werden. Seine Rivalen, die Taira, wollen dies verhindern. Der Spieler bekämpft die Taira mit Samurai-Kriegern, setzt sich gegen die Mörderbande der Ninjas zur Wehr, verbündet sich mit anderen Clans und zieht schließlich in die entscheidende Schlacht. Wird er der Shogun sein?

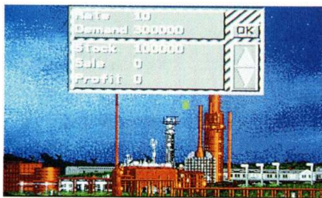
Streng vertraulich! Magnetic Scrolls hat **„Myth“** speziell für den Club Official Secrets entwickelt. Wer „Myth“ erwerben will, muß Club-Mitglied sein. Der Mitgliedsbeitrag beträgt



vier englische Pfund. Der Special Service des Clubs bietet das Clubmagazin Confidential (erscheint alle zwei Monate), das Adventure „Gnome Ranger“ und einen Telefonservice für Tricks und Tips. Jeder Einsender von Karten, Tips oder Lösungen hat die Chance, Hardware im Wert von 500 Pfund zu gewinnen. Club-Mitglieder erhalten Rabatt auf Spiele, die vom Club angeboten werden. Näheres bei John Trevillian, Official Secrets, U.K., Tel. 00 44-2 79-72 65 41. ▶

Kurz notiert:

In **„Oil Imperium"** von Reline hat's der Spieler besser als J.R. Ewing. Denn der hat alle naselang Ärger, weil er seine Konkurrenten auf die miese Tour ausschaltet. Der Spieler kann zeigen, daß man im Ölgeschäft auch mit fairen Mitteln zu etwas kommt — er hat dennoch die Möglichkeit, es im Stil eines



J.R. zu versuchen. Im Wettstreit um das Öl-Monopol spielt man gegen den Computer oder läßt bis zu drei weitere Spieler einsteigen. Gute Taktiken sind



vonnöten. So gewinnträchtig wie möglich wird gekauft, verkauft, sabotiert, fusioniert, integriert und spioniert. „Oil Imperium" ist ein Wirtschafts-Strategiespiel mit Actionelementen. Es kostet 60 Mark (Rushware).

Auf der Winter-CES in Las Vegas stellte Softgold **„Danger Freak"** vor. Der Spieler versucht als Stuntman ins Filmgeschäft einzusteigen. Mit halsbrecherischen Stunts muß er die Filmbosse begeistern. Je haarsträubender die Vorführung, desto besser sind die Chancen einer Anstellung. „Danger Freak" umfaßt drei Spielteile, die durch Zwischensequenzen verbunden sind. Im ersten Teil geht's auf den Highway to Hell, wo der Stuntman seine Knochen auf Hindernisfahrten und bei Kletterpartien auf fahrenden Autos zusammenhalten muß. Als nächstes muß man die haiverseuchte Bodega Bay umrunden, auf ein U-Boot klettern und sich von einem Hubschrauber auflesen lassen. Im letzten Teil überfliegt man eine Großstadt mit dem sogenannten Bond-Mobil. Dabei trifft man auf Damen, die einen unübertrefflichen Stunt zu bieten haben: sie reiten auf Besen durch die Luft und stören den eindrucksvollen Kunstflug des Stuntman — glücklicherweise hat der einen Fallschirm dabei.



In **„Realm of the Trolls"** brauchen freundliche Elfen Hilfe. Unverschämte Trolle haben sich am Elfenschatz vergriffen und ihn in 200 verschiedenen Trollgrotten versteckt. Lästige



Monster, böse Zauberer und natürlich die Trolle selber hüten die Beute wie ihren Augapfel. Im Fantasy-Rollenspiel soll der Elf Dirillon vom Spieler in die Trollwelt geführt werden und den Schatz suchen. Dabei muß er sich vor Wächtern, Falltüren und Blitzen in acht nehmen.

Der Planet **„Denaris"** war einst ein Paradies. Seine Bewohner ließen primitive Maschinen arbeiten und beschäftigten sich mit wissenschaftlichen Studien.



Dabei entwickelten sie Maschinen, die ihnen an Intelligenz gleichkamen und die sich folglich zu autarken Wesen entwikkelten. Machtstreben ersetzte den einstigen Arbeitseifer. Das Leben der Ureinwohner wurde unerträglich. Sie versuchten sich der Plage durch eine Atombombe zu entledigen, was zum kollektiven Beinahe-Selbstmord führte. Die eigenwilligen Roboter krochen jedoch aus der radioaktiven Schlacke und wurden halb Maschine, halb Organismus. Die überlebenden Denarier haben einen letzten Trumpf. Mit einem Kampfschiff schicken sie den Spieler gegen die machtgierigen Kreaturen.

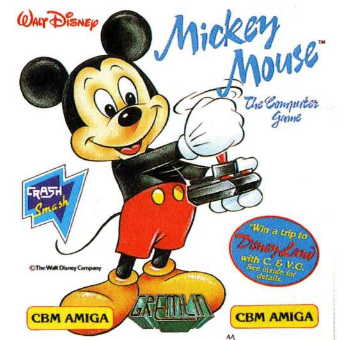
Hollywood und schöne Frauen — diese Begriffe gehören zusammen. **„Hollywood Poker Pro"** will das auf's neue belegen. Reline Software erfreut damit Pokerspieler und Voyeure. Dank einer Zoom-Funktion können digitalisierte Damen bis zur kleinsten Sommersprosse betrachtet werden. Das Pokerspiel kostet 60 Mark, das Hinschauen ist gratis.



Advanced Gravis aus Kanada stellt einen Joystick für hohe Ansprüche vor. In fünf Jahren Entwicklungszeit brachte man Bemerkenswertes zustande. Das Steuerungsgerät hat einen schaumbeschichteten Pistolengriff mit aufgesetztem Feuerknopf, drei von einander unabhängige Feuerknöpfe mit Mikroschaltern und acht Reaktionsstufen. Der Hersteller gibt ein Jahr Garantie.



Disneyland ist in Gefahr! Der König einer Meute Ungeheuer hat Merlins Zauberstab gestohlen. Schlimmer noch: Er hat ihn in vier Teile zerbrochen und die Bruchstücke in vier Türmen der Disneyburg versteckt. **„Mickey Mouse"** soll sich nun in die Burg wagen und all die gräßlichen Hexen und Monster besiegen, die die Bruchstücke des Zauberstabes bewachen. „Mickey Mouse" kommt von



Gremlin Graphics und wird von Ariolasoft vertrieben; 65 Mark. Berichtigung: In Ausgabe 1/89 gaben wir Rushware als Vertreiber für das Spiel **„Corruption"** an. Es wird jedoch von Ariolasoft vertrieben.

Delphine Software in Paris arbeitet an **„Bio Challenge"**, einem Roboterspiel mit 320 Farben, kunstvollen Grafiken und hervorragender Begleitmusik.



Paul Cuisset, der „Space Harrier“ programmierte, und der Grafiker Denis Mercier sind die Entwickler. **„Castle Warrior“**



stammt von Pierre Faure und Emmanuel le Coz. Die Grafiken wurden mit DPaint in 32 Farben angefertigt. Die musikalische Untermalung besteht hier nicht aus zusammengestükkelten Digi-Takten; Jean Baudlot, der auch für den eleganten Langweiler Richard Clayderman komponiert, hat für dieses Spiel eine achtminütige Komposition geschaffen. Die Spielspannung wird dadurch erhöht, daß auch musikalisch eine Spannung aufgebaut wird. Zum Komponieren verwendete Baudlot den „Ultimate Soundtracker“ von Karsten Obarski.



„Zak McKracken“, der rasende Reporter, recherchiert die Jahrhundert-Story. Er hat herausgefunden, daß Außerirdische das Telefonnetz manipulieren und die Menschheit per Draht verdummen. Während in aller Welt der IQ sinkt, sucht Zak nach den Übeltätern. Nachforschungen führen ihn nach Kalifornien, Ägypten, Eng-



land, Zaire, Mexico, in den Himalaya, auf die Bermudas und sogar zum Mars. Dabei hätte er bloß einmal unter seinen Schlafzimmerteppich schauen müssen. Dort ist eine Falltür versteckt. Sie führt zur Verdummungszentrale. Armer



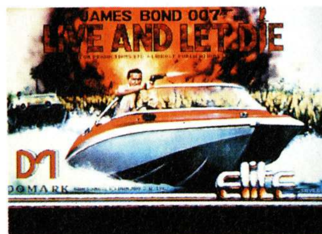
Zak! Er sucht und forscht und wird dabei selber immer dümmer. Der Spieler sieht dies im Menü: eine Aktions-Möglichkeit nach der anderen verschwindet.

Grafische Gestaltung und Animation erinnern an „Leisure Suit Larry“. Auch in Witz und Komplexität kann Zak es mit Larry aufnehmen.

James Bond 007 darf wieder „Leben und sterben lassen“. Im geheimen Auftrag Ihrer Majestät jagt er den Anführer einer Drogenbande. Das ganze läuft auf eine Geschicklichkeitfahrt mit einem Rennboot hinaus. Es geht durch Dschungel, zum Nordpol und durch die Sahara. Während Bond über die Gischt braust, muß er sich in acht



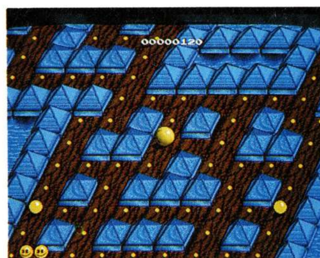
nehmen vor Minen, Sturzbomben, U-Booten und Hubschraubern. Für 80 Mark bei Bomico.



In **„Cosmic Bouncer“** wird ein simpler Tennisball durch kosmische Strahlen lebendig. Der kleine Ball tritt eine Reise durch's Weltall an. Hüpfend und springend eilt er über sich unberechenbar verändernde Wege. Für 70 Mark bei Casablanca und Leisuresoft.



Pacman in drei Dimensionen und mit fantastischer Sprungkraft — das ist **„Pac-Mania“**. Auf 22 Spielebenen ärgert das verfressene Kugelwesen sechs freche Gespenster. Vertreiber ist Ariolasoft.



Bezugsquellen und Vertreiber:

Amigaland
Hohenwaldstr. 26
6374 Steinbach

Ariolasoft GmbH
Hauptstr. 70
4835 Rietberg 2

Atlantis Soft- und Hardware
Dunantstr. 53
5030 Hürth 8

Bomico
Elbinger Straße 1
6000 Frankfurt 90

Casablanca GmbH
Nehringscamp
4630 Bochum 5

DTM
Bornhofweg 5
6200 Wiesbaden

GTI
Zimmersmühlenweg 73
6370 Oberursel

Leisuresoft
Industriestr. 23
4709 Bergkamen 5

Peksoft
Müllerstr. 44
8000 München 2

Profisoft
Sutthauser Str. 50/52
4500 Osnabrück

Rushware
Bruchweg 128-132
4044 Kaarst 2

Softgold
Hansaallee 201
4000 Düsseldorf 11

Softwareversand Müller
Dorfstr. 1
8852 Rain-Unterspeiching

technicSupport
Bundesallee 36-37
1000 Berlin 31

Hybris

Eulen schießen scharf

Raumstation „Hybris" startet in den Orbit des Planeten Jurica. Außerirdische Wesen sollen die dortige Kolonie erobert haben. Um sich Klarheit über die Situation zu verschaffen, schickt der oberste Befehlshaber seine verwegensten Piloten, J. P. Maverick und K. Lovett, zum Erkundungsflug.

Was die beiden dort erleben, ist wahrscheinlich die größte Knallerei, die jemals auf dem Amiga stattgefunden hat. Immer neue Formationen feindlicher Aliens fliegen dem kleinen Scout-Schiff entgegen. Ständig greifen eulenähnliche Wesen an, und Geschosse sausen aus den Geschützbunkern des Planeten. Während man die Bunker im Anfangslevel noch aus sicherer Distanz sprengen kann, wird es im weiteren Spielverlauf immer riskanter. Die merkwürdigen Eulen feuern intelligente Raketen, die sich ihr Ziel selber suchen. Bei derartigem Kugelhagel aus der Luft und vom Boden erscheint es nur gerecht, daß das Scout-Schiff mit vielen verschiedenen Superwaffen ausgerüstet ist. Deren Schlagkraft und Schußgeschwindigkeit erhöht sich im gleichen Maße, wie feindliche Schiffe und Wesen vernichtet werden. Sollte es trotzdem einmal happig werden, so läßt sich das Schiff mit einer schnellen Joystickdrehung in eine Art Kampfrakete mit Anhängsel verwandeln. Diese Verstärkung ist äußerst hilfreich, wenn es darum geht, die beiden Megamonster jedes Levels zu besiegen. Falls die drei vorhandenen Raumschiffe nicht zum Abschluß eines erfolgreichen Erkundungsflugs ausreichen, bietet „Hybris" die Continue-Option, die es auch weniger geschickten Ballerfreunden ermöglicht, den Planeten Jurica besser kennenzulernen.

„Hybris" von Discovery Software braucht den Vergleich mit erfolgreichen Ballerautomaten der Spielhallen-Szene nicht zu scheuen. An gutem Sound und guter Grafik mangelt es nicht. Das einzige, was bei „Hybris" fehlt, ist der Schlitz zum Geldeinwerfen.

(mms)



Luftkampf über HighTech-Oasen

Preis: 50 Mark
Hersteller: Discovery Software
Vertrieb: Casablanca, Atlantis

Spaß	10
Grafik	9
Sound	10
Idee	7
Schwierigkeit	8

Hellfire Attack

Kampfhummel im Einsatz

Super Cobra ist ein bis unter die Rotorblätter bewaffneter Hubschrauber. Mit diesem martialischen Fluggerät startet der Spieler Richtung Feindesland. Tag und Nacht knattert er über Wälder, Städte und Meeresbuchten und späht aus nach gegnerischen Flugzeugen und Hubschraubern. Landschaften und Details wie Häuser und Bäume werden schnell auf dem Bildschirm aufgebaut. Die Grafik ist allerdings so wackelig, als würde der Monitor gerade von einem Erdbeben erschüttert. Da! Am Horizont erscheinen feindliche Flieger — erst ganz klein, dann halbwegs kenntlich, schließlich nahe und bedrohlich. So entsteht der Eindruck von Weite und Räumlichkeit. Aber es bleibt wenig Zeit, die Illusion des Raumes zu genießen. Der Spieler muß sich den Feinden stellen. Der Feuerknopf läuft heiß. Unentwegt wird mit den Bordkanonen geschossen. Per Joystick bewegt der Spieler ein Fadenkreuz über den Bildschirm und visiert die Angreifer an. Ein Druck auf den Feuerknopf startet die Lenkflugrakete, die dann automatisch auf das Zielobjekt zurast. Damit allein wird der Kampf jedoch nicht entschieden. Denn während die Rakete den vorgegebenen Punkt anfliegt, hat der Feind genügend Zeit, sich mit genauso modernen, zielsuchenden Raketen für den Angriff zu revanchieren. Unser Pilot muß also eine Ausweichstrategie entwickeln. Nach einigen Game overs werden Gesetzmäßigkeiten offensichtlich; die Erfahrung zeigt, daß man den gegnerischen Geschossen nur dann entkommt, wenn man ganz tief am rechten oder linken Bildschirmrand entlangfliegt. Schießen, ausweichen, die Landschaft betrachten — damit ist die spielerische Originalität erschöpft. Zwar ändern sich die Fluginformationen des Gegners ständig, auch werden immer neue Landschaften und gleichzeitig schwierigere Level geboten, doch ist das Zielen mit dem Fadenkreuz auf die Dauer recht öd. Selbst die anfänglich aufregenden Ausweichmanöver geraten zur Routine. Martechs Ballerspiel ist deshalb ein Spaß für wenige Stunden.

(C. Borgmeier)



Martialisches Feuerwerk über lieblicher Landschaft.

Preis: 70 Mark
Hersteller: Martech
Vertrieb: Leisuresoft, Rushware

Spaß	7
Grafik	7
Sound	7
Idee	4
Schwierigkeit	6

Wolff's Computer Service

HOTLINE 04351/82295

3,5" 2DD No Name Disk	10St 22,50DM 100St. 220,-DM
5,25" 2D No Name Disk	10St. 6,90DM 100St. 65,-DM
MOUSE PAD (antistatische Mausunterlage)	St. NUR 8,50DM
MEDIA 3,5" Schubfach Diskbox	1St. 37,-DM 2St. 70,-DM
NEC P 2200 24 Nadel Matrixdrucker	St. 839,-DM
STAR LC-24-10 24 Nadel Matrixdrucker	St. 904,-DM
DATAPHON s 21 d-2 Akustikkoppler AMIGA/ATARI	SET 335,-DM

VORFÜHRGERÄTE – SERVICE

Ständig über 25 verschiedene Computer, Monitore und Drucker als Vorführgeräte zu fairen Preisen

AMIGA-PUBLIC-DOMAIN-SERVICE

Jede 3,5" PD-Disk inkl. Diskette nur 4,50 DM
Jede 5,25" PD-Disk inkl. Diskette nur 3,50 DM
Kostenlose PD-Information anfordern.

Fordern Sie unseren kostenlosen Gesamtkatalog mit über 900 verschiedenen Artikeln an bei:
WOLFF's COMPUTER SERVICE
Am Ort 6 / Postfach 1455
2330 Eckernförde
Tel.: 04351 / 82295

20 MB auf Floppy sichern

? ? ? ? ?

mit dem Sicherungsprogramm
cpio - Amiga
kein Problem!

Sichert 20 MB in 30 Minuten!
Sichert fortlaufend auf mehrere Disketten!
Sichert Dateien mit mehr als 880 KB!
Ein Programm aus Deutschland mit deutscher Anleitung und deutschem Dialog!

Preis: DM 79,- zuzügl. Versand

Beschreibung gegen Rückporto bei
Kurt Harders
Systemanalyse und Beratung
Angermunder Straße 252
4100 Duisburg 29
Tel. 0203/765377

Public Domain

Fish
Faug
TBAG
Auge
Panorama

AMIGA

Tornado
Kickstart
Taifun
u.s.w.

alles auf 2DD-Disk

bis	9 St.	**3,95 DM**
ab	10 St.	**3,80 DM**
ab	20 St.	**3,70 DM**
ab	30 St.	**3,60 DM**
ab	50 St.	**3,50 DM**
ab	100 St.	**2,99 DM**

Leerdisketten 3,5" 2DD ab **2,24 DM**
Info-Liste gegen Rückporto!

Versand nur gegen Vorkasse (Scheck) bzw. Nachnahme

Peter Keim
Vogelsanger Str. 34
5000 Köln 30

Telefon:
0221/
520765

Achtung!

1. Software darf nur vom Berechtigten vervielfältigt und verbreitet werden. Originalprogramme sind am Originalaufkleber des Datenträgers (Diskette oder Kassette) und ggf. am Copyright-Vermerk zu erkennen. Auch befinden sie sich normalerweise in der Originalverpackung. Wir empfehlen Käufern dringend, darauf zu achten, daß sie tatsächlich Originalprogramme erwerben.
2. Es verstößt gegen das Urheberrecht bzw. das Gesetz gegen den unlauteren Wettbewerb, Raubkopien herzustellen oder zu verbreiten. Verstöße können zivilrechtlich und ggf. auch strafrechtlich geahndet werden. Käufern von Raubkopien steht kein Nutzungsrecht zu. Sie müssen mit Unterlassungsverfahren und ggf. mit sofortiger Beschlagnahme der erworbenen Raubkopien rechnen.
   Wir erwarten von unseren Inserenten und Lesern, daß sie **keine** Raubkopien verbreiten, anbieten oder erwerben.
3. Anzeigen, die vermuten lassen, daß mit Raubkopien gehandelt wird, werden nicht veröffentlicht.

PROFIBU

Finanzbuchhaltung
und

PROFAKT

Fakturierung

Ausgereifte, doppelte Finanzbuchhaltung und abgestimmte Fakturierung für
AMIGA oder **MS/DOS**

PROFIBU: Frei definierbarer Kontenrahmen, schnelle und einfache Handhabung, autom. Abschlußmöglichkeit, Steuersplitting, Storno, Skonto, umfangreiche Auswertungen und Ausdrucke.

PROFAKT: Artikel, Lager, Kunden, Lieferanten, Sofortfakturierung, Bestellung, Gutschrift, Angebot, Lieferschein, Rechnung, Mahnung, Offene Posten, umfangreiche Auswertungslisten, Daten an PROFIBU.

Bitte fordern Sie Informationsmaterial an!

Entwicklung und Vertrieb:
Sven Tietjen
Rigaweg 1, 3300 Braunschweig, 05309/1466

Funkcenter Mitte GmbH

Klosterstr. 130 · 4000 Düsseldorf 1
Tel. 0211/362522 · Fax 0211/360195

Forms in Flight	169,—
Videoscape 3D	348,—
Page Flipper	98,—
DeLuxe Paint II	228,—
DeLuxe Video 1.2	228,—
DeLuxe Print	148,—
Aegis Audiomaster	148,—
Aegis Diga	178,—
Marauder II	89,—
AC Basic Compiler	348,—
DOS 2 DOS	128,—
Amiga Tools	39,—
GFA Basic	178,—

900 Public-Domain-Disketten für AMIGA!
2 Katalogdisketten gegen 5,- Briefmarken oder Schein anfordern.

Mailbox 24 Std. ONLINE, 0211/360104 8,N,1

Wolff's Computer Service

HALT 04351/82295 — STOP 04351/82295

WIR TAUSCHEN

100 St. 5,25" 2D No Name
Disk vollgestopft mit PD-Software
inklusive
5,25" COMBITEC
DISKLAUFWERK
inklusive
Diskbox für 100 Disk
gegen
4 2 9,--DM
IST DAS NICHTS ?!?

Fordern Sie auch unseren umfangreichen Gesamtkatalog sowie unsere PD-Software-Infos an
HOTLINE 04351/82295

Postfach 1455 Am Ort 6 2330 Eckernförde Tel. 04351/82295

STOP 04351/82295 — HALT 04351/82295

COMPUTER DIGITAL-TECHNIK

Die Commodore-Fachwerkstatt
Verkauf & Versand von Ersatzteilen
Neu-Anlagen
Reparatur-Express-Dienst
in der BRD
zu 95 % innerhalb einer Woche

Fa. HORST WITTE
Urbanstraße 132,
1000 BERLIN 61
Telefon 0 30/6 93 10 23

COMPUTER-BÖRSEN

25. + 26. 2. 89
4154 Krefeld-Tönisvorst Marienheim

4. + 5. 3. 89
5138 Heinsberg Festhalle Oberbruch

11. + 12. 3. 89
5020 Frechen bei Köln Schützenhaus

18. + 19. 3. 89
4300 Essen-Mitte Kolpinghaus

25. — 27. 3. 89
5603 Wülfrath Stadthalle

1. + 2. 4. 89
4630 Bochum-Wattenscheid Stadthalle

8. + 9. 4. 89
5600 Wuppertal Stadthalle

★★★

Jeder kann teilnehmen, Info:
02845/27260 (11 – 22 Uhr)

Zählt für Sie jeder Millimeter? Dann liegen Sie hier richtig mit Ihrem Angebot. Nutzen Sie diese Rubrik, um Ihre gewerbliche oder private Kleinanzeige zu plazieren. Wir berechnen DM 7,— zzgl ges. MwSt. pro Zeile für gewerbliche Anzeigen (1 Zeile = 33 Anschläge) und DM 5,— pro 5 Zeilen für private Anzeigen.

Und so wird es gemacht! Schicken Sie uns Ihren Text mit der genauen Adresse und nennen Sie uns die entsprechende Rubrik, unter der Ihre Anzeige erscheinen soll. Weitere Informationen geben wir Ihnen aber auch gerne unter Tel. 089/3 60 86-201.

GEWERBLICHE KLEINANZEIGEN

CHEMIE-SOFTWARE
yMolekül, das Programm zur graphischen Darstellung und Bearbeitung von Molekülen. Info von Cornelia Schmidt, Postf. 20 02 38, 1000 Berlin 20.

Haben Sie noch unveröffentlichte, selbsterstellte Programme in Ihrer Schublade? Hardware-Bastelei? Autoren für ein Amiga-Sammelwerk gesucht! Schneider, Am Weinberg 46, 8301 Arth

Gratisliste über Public-Domain für Amiga anfordern bei F. Neuper, 8473 Pfreimd, Postfach 72

AMIGA Virus-Falle; Hardware 49,95, EPROM-Brenner für Amiga 249,00, Atari ST, IBM, C64/128 Hard- und Software-Katalog bei: Roßmöller, Meckenheim, 0 22 25/20 61

Public Domain für Amiga
Je 3.5″ nur 3 DM, je 5.25″ nur 1.50 DM
Gratisinfo bei Jürgen Niederauer
Ludwig-Hotter-Str. 32
8925 Marktoberdorf

PRIVATE KLEINANZEIGEN

1000,— DM Belohnung!
Wer kann mir Hinweise auf den oder die Programmierer bzw. Urheber des DASA 0.2 — (Byte Warrior Virus geben? Tel. 0 89/6 99 03 07

!Achtung!
Suche Tauschpartner für Amiga 500 3,5″ Disketten.
Haas Alexander, Süsterfeldstr. 70, 5100 Aachen

Verkaufe „Kniffel"-Spiel für Amiga. Alle Spielregeln werden berücksichtigt. Mit Disk und Porto für DM 10,—. Schreibt an: Michael Ellerich, Traunsteinerstr. 4, 8220 Traunstein. Antwort innerhalb 1 Woche

Mach ne schnelle Mark! 100 erprobte Jobs. Diese Toncassette enthält 100 attraktive Möglichkeiten, mit denen junge Leute Geld verdienen können. DM 19,80. Fa. Autex, Molberger Str. 66, 4590 Cloppenburg; T.: 0 44 71-35 63

Tausche neueste Amiga-Software (mögl. 1000er). Disks an: Christian Hoffstadt, Bergstr. 58, 5942 Kirchhundem 5. Suche auch Programmierer!

Verkaufe Jagd auf Roter Oktober (35—25 DM) und Sentinel (30—20 DM) oder tausche. Beides Originale + Anleitungen. Call: Tilman Läppchen, Leipziger Str. 8, 6834 Ketsch, Tel: 06202/61860

Falls Sie Ihren Amiga auf professionelle Weise nutzen, sei es im DTP-, DTVideo-, Büro-, Grafik-, Textverarbeitungs-Bereich, sollten Sie sich sofort mit uns in Verbindung setzen. Wir sind der erste Club für alle professionellen Amiga-Anwender. GPA heißt Anwenderunterstützung, Demo-Service (Liste anfordern), Kontakte, Ideenaustausch, usw. Und das alles kostenlos (keine Beiträge oder Verpflichtungen). Anschrift: GPA, Hüttnerstr. 24, CH-8824 Schönenberg.

ZEISIG IS' DA!
Tauschpartner für Public Domain gesucht. Bester Stoff für AMIGA und PC. **P. D. U. G.** Sektion Zeisig Pf 11 18, D-6464 Linsengericht. Info gegen 2 × 100 Pf.

SUCHE Statikprogramm für Amiga 500 Bemessung und Berechnung für 1- + 2-Familien-Häuser, komplett Holz + Stahl + Stahlbeton + Mauerwerksbauteile. Wenn möglich bitte ich um Info + Angebot.
Meckel Rolf, Berliner Str. 9, 6312 Laubach

AMIGA-ANFÄNGER! Vk. eig. Broschüre „Amiga von Anfang an„ — u. a. Infos über CLI, Workbench, Hard- und Softwareinstallation; viele Antworten auf Fragen, die **nicht** im Handbuch stehen. Vom Fan für den Fan. Preis: 10,— DM incl. Porto: J. Schmidt, Aumunder Heide 47, 2820 Bremen 70, Tel. 04 21/ 65 73 75. Info gratis.

Kleinanzeigen-Service

AMIGA WELT

Die Zeitschrift für den Amiga Anwender

Jede weitere Zeile nur DM 1,— Bitte deutlich schreiben. Zwischen den Worten jeweils ein Kästchen freilassen.

Ob 1, 2 oder 6 Zeilen — jede Kleinanzeige für nur DM 5,—

Achtung! Sie erhalten keine Rechnung. Chiffre-Anzeigen sind nicht möglich. Damit Ihre Anzeige veröffentlicht wird, legen Sie bitte den entsprechenden Betrag in bar oder als Verrechnungsscheck bei.

Gewünschte Rubrik
☐ Verkaufe
☐ Suche ☐ Tausche

Produktbereich
☐ Hardware ☐ Peripherie ☐ Sonstiges
☐ Software ☐ Disketten

Bitte ankreuzen!

Absender

Name Vorname

Straße, Hausnr./Postfach

PLZ Ort

Vorwahl Telefon

Datum Unterschrift (bei Minderjährigen Unterschrift d. ges. Vertreters)

Der Betrag von DM _____ liegt
☐ in bar/
☐ als Verrechnungsscheck bei
(Zutreffendes bitte ankreuzen).

CW-Publikationen
AmigaWelt — Anzeigenverkauf
Rheinstr. 28

D-8000 München 40

Ausgefüllten Bestellschein in ein Fensterkuvert stecken und gleich abschicken!

Chefredakteur: Manfred S. Schmidt (mss) (verantwortlich, Anschrift siehe Redaktion)
Redaktion: Uwe Knierim (uk), Mike Nickles (mn), Ute Bahn (ub)
Ständige Mitarbeiter: Ilse und Rudolf Wolf (iw, rw), Mathias M. Schmidt (mms)
Redaktionsassistentin: Siggi Pöschel (sp)
Gestaltung: Karin Wirth
Anschrift der Redaktion: CW-Publikationen (Amiga-Welt), Postfach 40 04 29, Rheinstraße 28, 8000 München 40, Telefon: 0 89/3 60 86-0, Telex: 5 215 350 comw d, Telekopierer: 0 89/3 60 86-1 09
Auslandsredaktionen:
Österreich: COMPUTERWELT Österreich, CW-Publikationen Verlagsgesellschaft mbH, Zieglergasse 6/1, A-1070 Wien, Tel.: (00 43) 222/93 05 08
Schweiz: COMPUTERWORLD Schweiz, Witikonerstr. 15, Postfach 253, CH-8030 Zürich, Tel.: (01)55 10 77, Telex: 816 710 cw cich
USA: CW-COMMUNICATIONS INC., 375 Cochituate Road, Box 9171, USA-Framingham, Mass. 01701, Tel.: 001/617/879 07 00, Tx.: 00230/951 153 computwrld fmh
Japan: COMPUTERWORLD/Japan, 1-19-7, Shintomi Chuoku, J-Tokyo 104, Tel.: 00 81/3/5 51 38 82
China: China COMPUTERWORLD, 74 LuGuGun Road, P.O. Box 750, RC-Beijing 100039, Tel.: 00 88/6/814-61 74
Verlagsrepräsentanten:
Großbritannien: Beere Hobson Ass., Euan C. Rose, 34 Warwick Road, Kenilworth, GB-Warwickshire CV8 1HE, Tel.: 00 44/926/51 24 24.
USA: CW International Marketing Services, Diana La Muraglia, 375 Cochituate Road, Box 880, USA-Framingham, Mass. 01701, Tel.: 001/617/879 07 00, Tx.: 00230/951 153 computwrld fmh
Taiwan: Alice Chen ACE Media, Agency Co., Ltd., P.O. Box 26-578, Taipei R.O.C., Tel. 021721-4302, Telex 14142 acegroup
Japan: Steven Yamada, Tokyo Representative Corp., Sanshin Kogyo Jimbocho 3F, Chiyoda-ku, Tokyo 101 Japan, Tel.: 230-4117, Tlx.: (781) J26860

Einsendungen: Manuskripte werden gerne von der Redaktion entgegengenommen. Die Zustimmung zum Abdruck und zur Vervielfältigung wird vorausgesetzt. Gleichzeitig versichert der Verfasser, daß die Einsendungen frei von Rechten Dritter sind und nicht bereits an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten wurden. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen. Das Urheberrecht für angenommene und veröffentlichte Manuskripte liegt ausschließlich bei der CW-Publikationen Verlagsgesellschaft mbH. Nachdruck sowie Vervielfältigungen aller Art oder sonstige Verwertung von Texten aus Publikationen des Verlags nur mit schriftlicher Genehmigung. Namentlich gekennzeichnete Fremdbeiträge geben nicht unbedingt die Meinung der Redaktion wieder.

Einzelheft-Bestellungen: Direkt beim Verlag, Abt. Vertrieb (Anschrift s. u.) gegen Verrechnungsscheck + DM 2,– Porto oder per Nachnahme sowie über den Buchhandel.
Festbezug: 6 Ausgaben per Postversand im Inland DM 72,—. Auslandspreis DM/sfr 84,—. Bestellung ebenfalls beim Verlag oder über den Buchhandel.
ISSN 0933-8616
Vertrieb Handelsauflage: MZV Moderner Zeitschriften Vertrieb GmbH, Breslauer Straße 5, 8057 Eching, Tel.: 089/31 90 06-0, Telex: 522 656.
Telefon-Durchwahl: 089/3 60 86 ( )
Vertriebsleitung: Brigitte Schleibinger (-155/-156)
Anzeigenleitung: Sylvia Stier (-161); (verantwortlich für Anzeigen, Anschrift siehe unter Anzeigen)
Anzeigenverkaufsleitung: Barbara Schönberger (-130)

Impressum

CW PUBLIKATIONEN

Ihre starke Verbindung zur Computerwelt

Amiga-Welt ist eine Publikation der IDG Communications, der Welt größter Herausgeber für computerbezogene Informationen. IDG Communications veröffentlicht über 90 Computer-Publikationen in 33 Ländern. Jeden Monat lesen vierzehn Millionen Menschen eine oder mehrere Publikationen von IDG Communications. Alle Publikationen sind dem IDG International News Service angeschlossen, der die neuesten Meldungen aus der Kommunikations- und Informationstechnologie für die Redaktionen bereithält. IDG Communications-Publikationen sind:
ARGENTINIEN: Computerworld Argentina; ASIEN: Communications World, Computerworld Hong Kong, Computerworld Malaysia, Computerworld Singapur, PC Review, Computerworld SE Asia; AUSTRALIEN: Computerworld Australia, Communications World, Australian PC World, Australian Macworld; BRASILIEN: Data News, PC Mundo, Micro Mundo; BUNDESREPUBLIK DEUTSCHLAND: Computerwoche, Information Management, PC Woche, PC Welt, CHILE: Informatica, Computacion Personal; DÄNEMARK: Computerworld Danmark, PC World Danmark; ENGLAND: Computer News, DEC Today, ICL Today, Lotus, PC Business World; FINNLAND: Tietoviikko, Mikro; FRANKREICH: Le Monde Informatique, Distributique, Info PC, Telecoms International; GRIECHENLAND: Micro and Computer Age; INDIEN: Dataquest; ISRAEL: People & Computers Weekly, People & Computers Biweekly; ITALIEN: Computerworld Italia; JAPAN: Computerworld Japan; KANADA: Computer Data; MEXICO: Computerworld Mexico; NIEDERLANDE: Computerworld Nederland, PC World Benelux; NEUSEELAND: Computerworld New Zealand; NORWEGEN: Computerworld Norge, PC World Norge; ÖSTERREICH: Computerwelt Österreich; SAUDI ARABIEN: Arabian Computer News; SÜD-KOREA: Computerworld Korea, PC World Korea; SPANIEN: Cimworld, Computerworld Espana, Commodore World, PC World Espana, Communicaciones World, Informatica Industrial; SCHWEDEN: Computer Sweden, MikroDatorn, Svenska PC World; SCHWEIZ: Computerworld Schweiz; UNGARN: Computerworld SZT, PC Mikrovilag; VENEZUELA: Computerworld Venezuela; VEREINIGTE STAATEN VON AMERIKA: Amiga World, CD-ROM Review, CIO, Computer Currents, Computerworld, Computers in Science, Digital News, Federal Computer Week, 80 Micro, Focus Publications, InCider, Infoworld, Macintosh Today, Macworld, Computer + Software News (Micro Marketworld/Lebhar-Friedman), Network World, PC World, Portable Computer Review, Publish!, PC Resource, Run, Windows; VOLKSREPUBLIK CHINA: China Computerworld, China Computerworld Monthly

IDG
An IDG Communications Publication

Anzeigendisposition: Ursel Sauter (-126)
Anschrift für Anzeigen und Vertrieb: Postfach 400 429, Rheinstraße 28, 8000 München 40, Telefon: 0 89/3 60 86-0, Telex: 5 215 350 comw d, Telekopierer 089/3 60 86-1 09
Disposition: Rainer Oberländer (Leitung), Stefan Liba
Druck und Beilagen: Carl Gerber Grafische Betriebe, Muthmannstraße 4, 8000 München 45, Tel.: 089/3 23 93-233 (Anschrift für Beilagen)
Zahlungsmöglichkeiten: Bayerische Vereinsbank, BLZ 700 202 70, Konto-Nr. 116 000, Pschk. München 97 40-800
Erfüllungsort, Gerichtsstand: München
Verlag: CW-Publikationen Verlagsgesellschaft mbH, Rheinstraße 28, 8000 München 40, Telefon: 089/3 60 86- 0, Telex: 5 215 350 comw d, Telekopierer: 089/3 60 86-109
Art Director: Darinka Bratusa
Produktionsleitung: Heinz Zimmermann
Redaktionsdirektor: Dieter Eckbauer
Kaufm. Leitung: Jürgen T. Reischl
Geschäftsführer: Eckhard Utpadel, Walter Boyd, Patrick McGovern

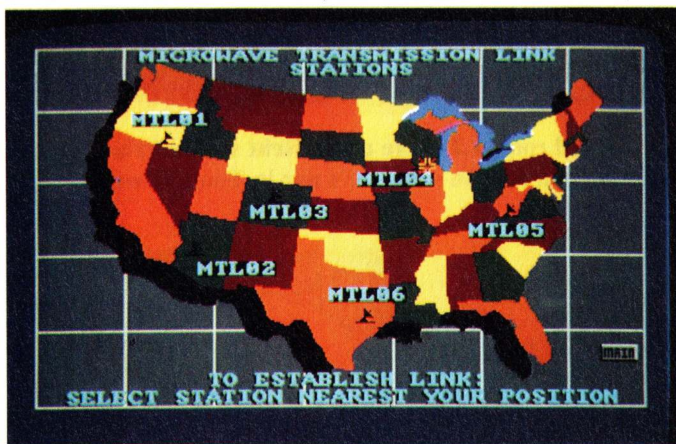
INSERENTENVERZEICHNIS

In dieser Ausgabe befindet sich eine Beilage des Kunden „technic Support", 1000 Berlin. Wir bitten um freundliche Beachtung.

Def Con 5

Missiles statt Argumente

„Def Con 5" basiert auf folgender Vorgeschichte: Russen und Amerikaner haben ihre diplomatischen Beziehungen abgebrochen. Eine militärische Auseinandersetzung ist zu befürchten. Die Supermächte berufen ihre Krisenstäbe ein; Sachverständige, von denen man annimmt, sie seien verantwortungsbewußt und zuverlässig, werden um Rat gefragt.
Die Simulation „Def Con 5" beginnt an diesem Punkt höchster politischer Gefahr. Vorsorglich bemannen die Amerikaner die SDI-Raumstation und setzen den Spieler als Kommandanten ein. Die Russen schicken Weltraumminen in den Orbit, um die Waffensysteme der SDI-Station zu zerstören. Ohnmächtig, was die Gegebenheiten auf der Erde betrifft, harren Kommandant und Besatzung der Ereignisse. Im Falle eines Übergriffs lautet ihr Befehl, ICBM-Angriffe (Intercontinental Ballistic Missile) auf die USA und ihre Verbündeten abzuwehren. Man findet keine politische Lösung. Die UdSSR löst den Erstschlag aus.
Sofort sendet der Kommandant ferngelenkte Roboter aus, die mit ihren Greifern Minenzünder demontieren sollen. Von dem Moment, in dem der Greifarm die Mine berührt, bleiben 30 Sekunden Zeit zum Entschärfen — oder bis zur Explosion. Nachdem alle Minen entschärft oder explodiert sind, beginnt die eigentliche Offensive. Russische Interkontinentalraketen sind gestartet worden. Zu deren Abwehr hat der Kommandant nun alle Waffen zur Verfügung, die auch für das SDI-Projekt geplant sind. Er startet eine Abwehrwaffe. Die beiden Raketen nähern sich. Gespannt beobachtet der Spieler die Flugkörper. Jeden Augenblick erwartet er den Knall einer Detonation. Aber nichts ist zu hören, denn im Orbit gibt es zu wenig Luft, als daß sich der Schall fortpflanzen könnte. In gespenstischer Stille explodieren die Raketen. Und weit unten sieht man den blauen Planeten und wird sich bewußt, wie nahe sein Untergang ist.
Hervorragende Grafik und Animation täuschen Realität vor; die schwarzweißmalerische Story zerstört diesen Eindruck.
(R. de Flers)



Falls SDI-Systeme versagen, wird die Bodenabwehr aktiviert.

Preis: 80 Mark
Hersteller: Microprose/Cosmi
Vertrieb: Rushware

Spaß	8
Grafik	10
Sound	keiner
Idee	6
Schwierigkeit	10

Motorbike Madness

Über Stock und Stein

In „Motorbike Madness" muß man mit einer sensiblen Enduro-Maschine in drei Minuten einen anspruchsvollen Geländeparcours bewältigen, und zwar ohne Sturz. Mit Riesensätzen springt die Maschine über Flüsse und kleinere Hindernisse, rattert über Autowracks und alte Reifen. Sobald zehn Prozent der Hindernisstrecke bewältigt sind, ist es höchste Zeit für eine Reparaturpause. Wie groß der Schaden am Motorrad ist, läßt sich anhand der Stürze feststellen. Die Schäden zu erkennen, heißt jedoch nicht, sie auch alle zu beheben, denn dazu reicht oft das Geld nicht. Ramponiert und in TÜV-feindlichem Zustand geht es dann auf den nächsten Streckenabschnitt. Hier muß man das Gleichgewicht halten können. Ein schmaler Baumstamm wird überquert; danach geht es über einen ausgedienten VW-Käfer zur Leiter, die den Enduro-Fahrer trockenen Reifens über den Fluß fahren läßt. Sollte der Fahrer bei all den Hindernissen, Fahnen und Bäumen einmal die Übersicht verlieren, so kann der Spielablauf mit der Taste „F" angehalten werden. Mit der „U"-Taste geht's weiter.
Es gibt mehrere Möglichkeiten, die Hindernisstrecke hinter sich zu bringen. Allerdings ist der einfachste Weg nicht immer der beste. Wer sämtliche Hindernisse umfährt, bekommt keine Punkte. Beim nächsten Reparaturstopp wird er es bitter bereuen, denn dort sind Punkte bares Geld. Wer kein Risiko scheut, erwirbt schnell die nötige Barschaft, um sein Motorrad entsprechend in Schuß zu halten. Aber auch ein pralles Portemonnaie schützt nicht vor der Disqualifikation nach dem siebenten Sturz. Ein Kompromiß zwischen Tollkühnheit und Sicherheit ist das beste Rezept, um bei „Motorbike Madness" mithalten und vielleicht gewinnen zu können.
Die Enduro ist schwierig zu handhaben. Zum einen reagiert die Maschine sensibel auf jede Joystick-Bewegung, zum anderen muß man Schwung und Geschwindigkeit vor Schanzensprüngen berechnen und mit der „wheelie"-Taste (Fire-Button) und den Bremsen umgehen können. Ist man damit erst einmal vertraut, ist kein Berg zu hoch und kein Sprung zu weit.
(mms)



Könner überqueren Wassergräben im Fluge.

Preis: 60 Mark
Hersteller: Mastertronic
Vertrieb: Rushware

Spaß	7
Grafik	6
Sound	4
Idee	7
Schwierigkeit	9

Roger Rabbit

Rächer der Enterbten

Im Kinospektakel „Roger Rabbit" wetteifern Comic-Figuren und Schauspieler um die Gunst des Publikums. Zu einer so aufwendigen und teuren Hollywood-Produktion darf ein Computerspiel nicht fehlen. Coktel Vision hat die Story jetzt versoftet. Folgendes ist passiert: Marvin, der greise Besitzer der Toonstadt, hat testamentarisch festgelegt, daß die Comic-Figuren — auch Toons genannt — die Stadt erben. Jetzt ist Marvin tot. Doch wo ist sein Testament? Die Toons müssen das Schriftstück unbedingt finden, denn anderenfalls würde Toonstadt in die Hände eines raffgierigen Richters fallen. Dieser Mann des Gesetzes ist so abgrundtief böse, daß er nicht einmal vor Mord zurückschrecken würde. Seine „Erbfeinde", die Comic-Figuren, lassen sich jedoch nur mit einer Suppe spezieller Rezeptur vernichten. Und ein solches Gebräu kocht der tückische Richter. Doch da tritt Roger Rabbit in Aktion. Der quirlige Comic-Hase macht sich auf die Suche nach dem Testament, rechnet mit dem Richter ab und auch mit dessen Schergen, den Wieseln. Das Spiel zum Film besteht aus drei verschiedenen Spielsequenzen.

Zunächst rast Roger mit seinem Spezialauto zum Ink & Paint Club. Auf dem Weg dorthin drohen Kollisionen mit Straßenbahnen und Autos. Was aber ein echter Hase ist, der überwindet diese Hindernisse mit seinem springenden Auto. Jeder Druck auf den Feuerknopf löst einen Hopser aus. Entgegenkommenden Autos muß Roger ausweichen. Vorsicht! Hin und wieder sind Pfützen mit der gefährlichen Suppe auf der Straße zu sehen. Auch diese Gefahr muß übersprungen werden, sonst ist eins der fünf Bildschirmleben ausgehaucht. Wenn Roger den Ink & Paint Club lebendig und wohlbehalten erreicht, wechselt das Szenario. Roger rennt und hoppelt nun so schnell er kann um die Clubtische und sammelt Zettel ein. Währenddessen geben Bühnenmusiker ein Stück zum Besten. Sobald sie ihr Gedudel beendet haben, muß Roger alle Papiere beisammen haben.

Erschwerend kommt hinzu daß zwei unermüdliche Pinguine ständig weitere Schriftstücke auf den Tischen verteilen. Eins davon ist das Testament des seligen Marvin. Wer dieses wichtige Papier gefunden hat, spielt in der dritten und letzten Sequenz weiter. Roger befindet sich jetzt in einer Scherzartikelfabrik. Mit Hilfe bestimmter Scherzartikel werden die Wiesel, die flinken Komplizen des üblen Richters unschädlich gemacht. Benutzt Roger nämlich die richtigen Scherzartikel, lachen sich die Wiesel kaputt — im wahrsten Sinne des Wortes. Nach den Wieseln ist der Richter dran. Der Sieg über den Bösewicht bedeutet die Rettung für die Toonstadt. Damit ist das Spiel gelöst.

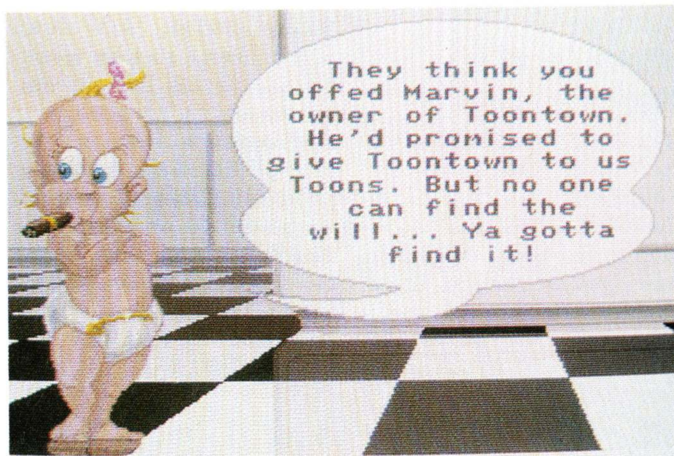
„Roger Rabbit" bietet farbenprächtige Grafiken und annähernd originalgetreue Cartoon-Figuren. Die Animation, speziell die der zweiten Spielsequenz, ist schnell und flüssig. Apropos flüssig: Besonders komisch inszeniert sind Rogers zaghafte Trinkversuche. Wer den Film kennt, weiß, was passiert. Kaum hat Roger am Alkohol genippt, schüttelt er sich vor Ekel und springt vor lauter Entsetzen bis an die Decke.

Was Grafik und Spielinhalt betrifft, ist mit „Roger Rabbit" ein guter Wurf gelungen. Der Sound ist jedoch nicht sonderlich gut. Ferner ist zu bemängeln, daß Besitzer eines einzigen Laufwerkes ständig die beiden Disketten wechseln müssen, ehe das Spiel überhaupt beginnt.

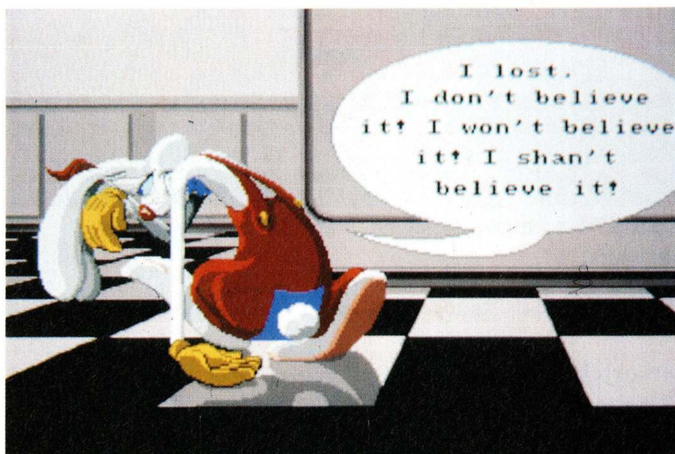
(C. Borgmeier)

Preis: 75 Mark
Hersteller: Coktel Vision
Vertrieb: Bomico

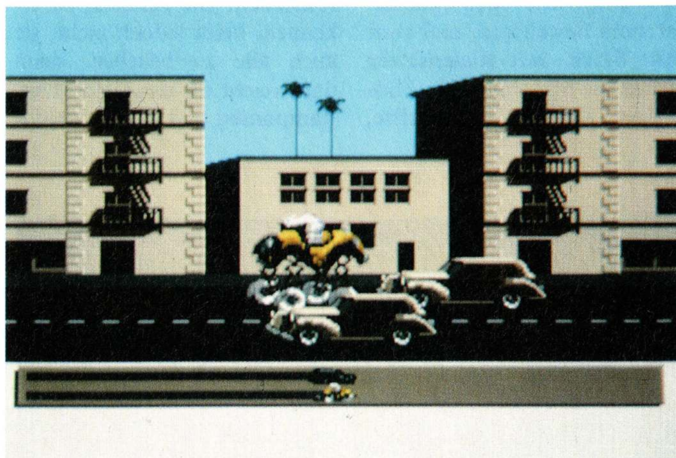
Spaß	8
Grafik	8
Sound	6
Idee	7
Schwierigkeit	8



Wickelkind raucht Zigarre und mischt sich in Erbauseinandersetzungen ein. So was gibt's nur in Toon-Town.



Langohr Roger kann den üblen Richter nicht austricksen. Mit hängenden Löffeln beklagt er sein Schicksal.



Roger muß als erster im Club sein. Wenn Autos im Weg stehen, fährt er die Sprungfedern aus und überwindet das Hindernis im Fluge.

Sword of Sodan

Reise durch das Zeitalter der Verzauberung

Fantasy-Abenteuer auf Datenträger zeichnen sich in aller Regel dadurch aus, daß sie hübsch simpel gut und böse unterscheiden. „Sword of Sodan" spielt in jener leicht überschaubaren Welt. Zoras, der schwarze Hexenmeister verkörpert das Böse. Daß er in einem Turm aus Menschenknochen haust, soll den Spieler ahnen lassen, wie gefährlich er als Gegner ist. Zu allem Unglück hat Zoras Macht über das Königreich des Nordens erlangt. Zwielichtiges Gesindel, das Gefolge des Hexenmeisters, treibt sich seitdem auf den Straßen herum. Es versucht den alten König gänzlich zu entmachten und seine beiden Kinder zu töten. Es gelingt dem König jedoch, den Prinz und die Prinzessin in Sicherheit zu bringen. Er selbst stirbt durch Zoras' Teufelsflamme.

Die Jahre vergehen. Brodan und Shardan, die Königskinder, wachsen heran. Beide sind inzwischen zu todesverachtenden Kämpfern ausgebildet worden. Doch nur einer von ihnen kann Rache nehmen, da es nur ein einziges Schwert gibt, dessen Streich für Zoras tödlich ist. Der Spieler wählt Kämpfer oder Kämpferin. In des Auserwählten Hände wird das sagenumwobene Schwert des Sodan gelegt. Dann geht es nach Craggamoore, der Hauptstadt des Reiches, in der Zoras ein grausames Regiment führt.

Halt, Fremder! Vier Torwachen verwehren dem Rächer Einlaß. Doch was sind vier Lanzen gegen die Kraft eines Zauberschwertes? Der Rächer springt mit einem mächtigen Satz auf die Wachen zu und streckt sie nieder. Nun schnell durch das Stadttor und über die Zugbrücke! Dabei ist Vorsicht geboten, denn spitze Metallpfähle durchbohren die Brükkenbohlen in regelmäßigen Abständen. Am Ende der Brücke lauert ein weiterer Schwerbewaffneter. Zu allem Überfluß versinken nun die Holzpfähle unter den Füßen des Rächers. Schwerverletzt, doch vom Haß weitergetrieben, erreicht er die Stadt. Auf dem Weg zum Schloß stellen sich zwei schlitzäugige Soldaten mit großen Beilen in den Weg. Dann ein Riese mit einer schweren Keule. Wegelagerer greifen an. Zwei Drachen mit sichelbewehrten Schwänzen brechen aus dem Wald und schlagen mit ihren tödlichen Waffen nach dem Rächer. Ein getöteter Drache ist als Skelett dargestellt. Auf einem Friedhof, den der Rächer überqueren muß, treiben feuerspeiende Zombies ihr Unwesen. Erst nachdem alle besiegt sind, darf der Held ins Schloß Craggamoore eintreten und weitere Beweise seiner Kühnheit liefern. Denn eine Schloßbesichtigung wird es nicht werden. Falltüren, spitze Pfähle, fliegende Ungeheuer und glühende Lava sind die Mittel, mit denen Zoras den Eindringling aufzuhalten sucht. Aber all seine Anstrengungen werden umsonst sein; der Spieler, inzwischen völlig von der Handlung gepackt, wird Zoras keine Chance lassen.

Grafik und Animation gaukeln eine recht überzeugende, lebendige kleine Welt vor. Die animierten Bobs, also Spielfiguren, sind ungewöhnlich groß. Der Spieler kann nicht umhin, sich in das interaktive Abenteuer zu stürzen und für die Gerechtigkeit zu kämpfen, zumal er – falls eine Speichererweiterung vorhanden ist – alle Schand- und Heldentaten in einem kleinen Film, sozusagen als RAM-Wiederholung, betrachten kann.

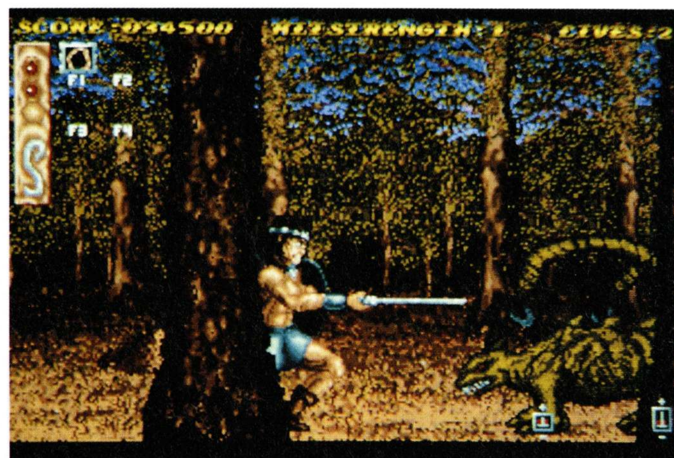
(mms)

Preis: 75 Mark
Hersteller: Discovery Software
Bezugsquelle: Atlantis, Amigaland

Spaß	10
Grafik	10
Sound	9
Idee	8
Schwierigkeit	9



Spieße drohen den Helden zu durchbohren. Ein Sprung rettet ihn. Doch da lauert schon die nächste Gefahr.



Glutäugige Drachen schlagen mit messerscharfen Haken nach dem Rächer.



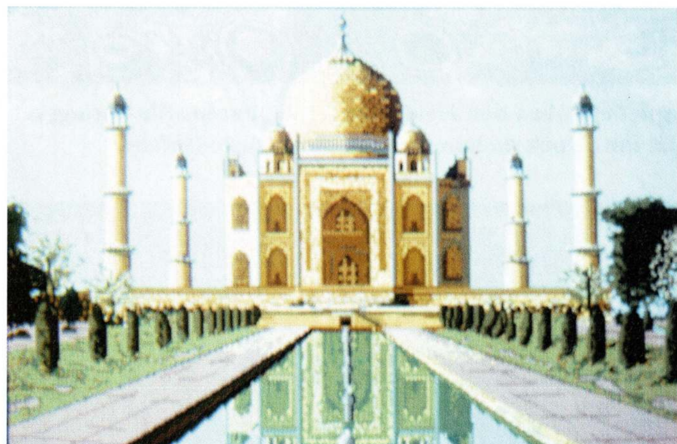
Der Kämpfer mit dem magischen Schutzschild ist schwer zu schlagen. Selbst gegenüber Königstöchtern zeigt er keine Milde.

Chronoquest

In die Zukunft via Steinzeit

Wie faszinierend wäre es doch, ohne Schranken durch Zeit und Raum zu fahren! So mag der Leser von H. G. Wells Novelle „Die Zeitmaschine“ ausrufen. Psygnosis erfüllt den Wunsch mit „Chronoquest“. Der Spieler schlüpft in die Rolle eines jungen Mannes, der zu Unrecht des Mordes an seinem Vater verdächtigt wird. Aus dem Abschiedsbrief des ahnungsvollen Vaters weiß der Verdächtigte, daß des Vaters Diener Richard den Mord begangen hat und ihn dafür zu liefern, muß Richard gefunden werden. Der ist allerdings mit einer Zeitmaschine ins Paris der Zukunft geflüchtet. Da ihm bekannt war, daß eine Zeitmaschine nach Benutzung immer an den Ausgangsort und in die Gegenwart zurückkehrt, mußte er mit Verfolgung rechnen. Deshalb zerbrach er die Programmkarte mit dem Steuerprogramm für die Zukunft und versteckte die vier Bruchstücke in der Vergangenheit. Nun ist es die Aufgabe des Spielers, die Zeitmaschine in dem verwinkelten Schloß zu finden. Hinweise sind rar, eine Anleitung gibt es nicht. Ist die Zeitmaschine entdeckt, so wird sie mit wenigen Handgriffen gestartet, und ab geht es! Blitzschnell werden vergangene Jahrhunderte durchreist, bis der erste Haltepunkt erreicht ist. Der Spieler befindet sich jetzt im Paläolithikum wo der tückische Richard ein Bruchstück jener Programmkarte versteckt hat, die die Zeitmaschine ins Paris des Jahres 2125 steuern kann. Es bleibt dem Spieler also nicht erspart, in vier Zeiten an vier Orten nach den Fragmenten der Karte zu suchen. Von der Steinzeit aus geht es direkt zum Anfang des 17. Jahrhunderts nach Indien, von dort mit Zwischenstation im Mexiko um 750 n. Chr. ins Ägypten des 12. Jahrhunderts. Jetzt endlich kann es ins Paris der Zukunft gehen. Und dann ist es nur noch eine Frage der Zeit, den Mörder zu stellen. Wie von einem Psygnosis-Spiel erwartet, sind die Grafiken ansprechend und gelungen. „Chronoquest“ ist benutzerfreundlich, in deutscher Sprache und ohne Texteingabe. „Chronoquest“ bietet Tüftlern und Hobby-Detektiven Stunden und Tage voller Spannung.

(R. de Flers)



Im 17. Jahrhundert war Tadsch Mahal noch ein Neubau.

Preis: 80 Mark
Hersteller: Mandarin/Terrific Software
Bezugsquelle: Müller

Spaß	7
Grafik	10
Sound	9
Idee	6
Schwieri	

Wizard Warz

Wer erbt Merlins Macht?

Es ist der Traum eines jeden Zauberlehrlings, so mächtig wie der gefürchtete Merlin zu werden. Dabei darf man allerdings nicht vergessen, daß selbst ein Merlin eine harte Lehrzeit hinter sich bringen mußte, bevor die Götter ihm unglaubliche Kräfte verliehen. Wie rauh der Weg zum Ruhm ist, bekommt der kleine Zauberlehrling aus „Wizard Warz“ am eigenen Leibe zu spüren. Drei höllisch schwere Aufgaben muß der ruhmesgierige Grünschnabel lösen, ehe er sich Meister nennen darf. Böse Geister überschwemmen den Bildschirm. Unserem Zauberlehrling bleibt nichts anderes übrig, als fleißig gegen diese Plage anzuzaubern. In der ersten Spielstufe geht der Schüler auf Wanderschaft und versucht, den zaubernden Unholden sechs gestohlene Gegenstände zu entreißen und zum rechtmäßigen Besitzer zurückzubringen. Das gibt zur Belohnung eine kraftspendende Wegzehrung.

Um einige Erfahrungen reicher, soll er seine Fortschritte im zweiten Level unter Beweis stellen. Dutzende Gegner stürzen sich auf den kleinen Zauberer und versuchen, ihn seiner körperlichen, geistigen und übernatürlichen Kräfte zu berauben. List und Vorsicht sind die geeigneten Mittel, sich selbst zu schützen, und einen Gegner nach dem anderen niederzuzaubern. Jeder Erfolg wird mit einem Zauberspruch aus dem Repertoire des Gegners belohnt. Auf diese Weise füllt sich das Zauberbuch des Lehrlings Seite um Seite. Ist auch das letzte Blatt mit Runen und Sprüchen bekritzelt, ist unser Held gewappnet für das Zauberfinale. Sieben Mitbewerber um die Nachfolge Merlins, des großen britannischen Meisters, erwarten ihn. Jeder der Sieben muß besiegt werden. Neben dem 7×7 Zentimeter großen Kampfplatz sieht man zur einen Seite ein Portrait des Zauberlehrlings und des jeweiligen Gegners. Auf kleinen Kräftemessern läßt sich die Veränderung der physischen, mentalen und übernatürlichen Kräfte ablesen. Daneben sind die aufgeschlagenen Zauberbücher abgebildet. Die darin enthaltenen Zaubersprüche dienen als Waffen. Ausgerüstet mit Tastatur und Joystick obliegt nun dem Spieler, den Nachfolger des edlen Merlin zu ermitteln.

(R. de Flers)



Die löwenköpfigen Zauberer haben keine mentalen Kräfte. Das ist die Chance für den Zauberlehrling.

Preis: 80 Mark
Hersteller: Canvas Software
Vertrieb: Rushware

Spaß	7
Grafik	7
Sound	2
Idee	6
Schwierigkeit	8

Fortsetzung von Seite 97

```
381  FOR a=0 TO 500:NEXT a                                      346
382  f$="Bitte geben Sie Punkt B an !"                          407
383  CALL
      SetWindowTitles(WINDOW(7),SADD(f$+CHR$(0)),0)             077
384  Test=MOUSE (0)                                             436
385  WHILE MOUSE(0)=0:WEND                                      108
386  ox2=MOUSE(1):oy2=MOUSE(2)                                  274
387  IF POINT(ox2,oy2)<0 THEN BEEP:GOTO aver                    411
388  IF oy2<oy THEN SWAP oy,oy2                                 556
389  IF ox2<ox THEN SWAP ox,ox2                                 725
390  f$="Zaubermeister c 1988 by NEUDELSOFT"                    929
391  CALL
      SetWindowTitles(WINDOW(7),SADD(f$+CHR$(0)),0)             079
392  fakx=(ox2-ox)/(mx2-mx)                                     917
393  faky=(oy2-oy)/(my2-my)                                     952
394  Test=-1:                                                   065
395  FOR a=mx TO mx2                                            521
396   Test=Test+1:c=-1                                          234
397   FOR b=my TO my2:c=c+1                                     554
398    LINE (ox+Test*fakx,oy+c*faky)-                           661
399 (ox+Test*fakx+fakx,oy+c*faky+faky),POINT(a,b),bf            244
400    NEXT b                                                   901
401  NEXT a                                                     053
402  GOTO warte                                                 545
403  ainv:                                                      481
404  FOR a=mx TO mx2                                            553
405   GET (a,my)-(a,my2),clip%                                  045
406   PUT (a,my),clip%,PRESET                                   770
407  NEXT a                                                     930
408  GOSUB Testen                                               725
409  GOTO warte                                                 449
410  Testen:                                                    454
411   IF (amx+amx2+amy+amy2)>0 THEN                             398
412    mx=amx:mx2=amx2:my=amy:my2=amy2                          132
413    amx=0:amx2=0:amy=0:amy2=0                                868
414   END IF                                                    330
415  RETURN                                                     978
416  akipp:                                                     657
417  haelfte=INT(my+(my2-my)/2)                                 105
418  Test=-1                                                    056
419  FOR a=my TO haelfte                                        823
420   Test=Test+1                                               504
421   GET (mx,a)-(mx2,a),clip%                                  826
422   GET (mx,my2-Test)-(mx2,my2-Test),tausch%                  612
423   PUT (mx,a),tausch%,PSET                                   053
424   PUT (mx,my2-Test),clip%,PSET                              978
425  NEXT a                                                     498
426  GOSUB Testen                                               037
427  GOTO warte                                                 377
428  aspi:                                                      852
429  haelfte=INT(mx+(mx2-mx)/2)                                 481
430  c=-1                                                       975
431  FOR a=mx TO haelfte                                        933
432   c=c+1                                                     320
433   GET (a,my)-(a,my2),clip%                                  996
434   GET (mx2-c,my)-(mx2-c,my2),tausch%                        172
435   PUT (a,my),tausch%,PSET                                   850
436   PUT (mx2-c,my),clip%,PSET                                 194
437  NEXT a                                                     866
438  GOSUB Testen                                               637
439  GOTO warte                                                 145
440  akop:                                                      534
441  f$="Bitte geben Sie die linke obere Ecke an !"             993
442  CALL
      SetWindowTitles(WINDOW(7),SADD(f$+CHR$(0)),0)             655
443  Test=MOUSE(0)                                              634
444  WHILE MOUSE(0)=0:WEND                                      010
445  ox=MOUSE(1):oy=MOUSE(2)                                    244
446  abst=mx2-mx                                                268
447   FOR b=0 TO abst                                           491
448    GET (mx+b,my)-(mx+b,my2),clip%                           141
449    PUT (ox+b,oy),clip%,PSET                                 504
450   NEXT b                                                    849
451  f$="Zaubermeister c 1988 by NEUDELSOFT"                    629
452  CALL
      SetWindowTitles(WINDOW(7),SADD(f$+CHR$(0)),0)             323
453  GOTO warte                                                 007
454  aloe:                                                      018
455  LINE (mx,my)-(mx2,my2),0,bf                                851
456  GOTO warte                                                 687
457  amark:                                                     465
458  f$="Bitte geben Sie Punkt A an !"                          593
459  CALL
      SetWindowTitles(WINDOW(7),SADD(f$+CHR$(0)),0)             663
460  Test=MOUSE(0)                                              610
461  WHILE MOUSE(0)=0:WEND                                      290
462  mx=MOUSE(1):my=MOUSE(2)                                    976
463  IF POINT(mx,my)<0 THEN BEEP:GOTO amark                     711
464  FOR a=1 TO 100:NEXT a                                      039
465  f$="Bitte geben Sie Punkt B an !"                          616
466  CALL
      SetWindowTitles(WINDOW(7),SADD(f$+CHR$(0)),0)             262
467  Test=MOUSE (0)                                             901
468  WHILE MOUSE(0)=0:WEND                                      665
469  mx2=MOUSE(1):my2=MOUSE(2)                                  423
470  IF POINT(mx2,my2)<0 THEN BEEP:GOTO amark                   730
471  IF my2<my THEN SWAP my,my2                                 915
472  IF mx2<mx THEN SWAP mx,mx2                                 396
473  f$="Zaubermeister c 1988 by NEUDELSOFT"                    060
474  CALL
      SetWindowTitles(WINDOW(7),SADD(f$+CHR$(0)),0)             250
475  GOTO warte                                                 438
476  farbarbeiten:                                              429
477  ON b GOTO
      hd,grau,rgb,schwimmen,kontr,psng,antik,zufall,fi
      l,sort                                                    296
478  sort:                                                      934
479  form=0                                                     403
480  WINDOW 5,"Sortieren",(10,0)-(310,40),0,1                   572
481  COLOR 0,1:CLS                                              724
482  LOCATE 2,5:PRINT ">":LOCATE 2,10:PRINT "<":LOCATE
      2,15:PRINT "OK"                                           423
483  LINE (30,5)-(55,25),0,b                                    957
484  LINE (70,5)-(95,25),0,b                                    931
485  LINE (110,5)-(135,25),0,b                                  567
486  SOUND 880,5,200,0                                          499
487  sowarte:                                                   094
488  Test=MOUSE(0)                                              781
489  WHILE MOUSE(0)=0:WEND                                      257
490  x=MOUSE(1):y=MOUSE(2)                                      657
491  IF y>5 AND y<25 THEN                                       526
492   IF x>30 AND x<55 THEN form=1                              168
493   IF x>70 AND x<95 THEN form=2                              403
494   IF x>110 AND x<135 THEN WINDOW CLOSE 5:GOTO
      warte                                                     426
495  END IF                                                     456
496  IF form=0 THEN GOTO sowarte                                037
497  WINDOW CLOSE 5                                             012
498  FOR a=0 TO 31                                              760
499   pointer(a)=a                                              197
500  NEXT a                                                     885
501  FOR z=0 TO 31                                              502
502   FOR a=0 TO 30                                             313
503    summe=Farben%(a,0)+Farben%(a,1)+Farben%(a,2)             705
504    c=Farben%(a+1,0)+Farben%(a+1,1)+Farben%(a+1,2)           333
505    IF form=1 THEN SWAP c,summe                              974
506    IF c<summe THEN                                          902
507     SWAP Farben%(a,0),Farben%(a+1,0)                        491
508     SWAP Farben%(a,1),Farben%(a+1,1)                        822
509     SWAP Farben%(a,2),Farben%(a+1,2)                        567
510     FOR b=0 TO 31                                           688
511      IF pointer(b)=a THEN pointer(b)=a+1:b=b+1              289
512      IF pointer(b)=a+1 THEN pointer(b)=a                    487
513     NEXT b                                                  264
514    END IF                                                   182
515   NEXT a                                                    694
516  NEXT z                                                     183
517  FOR a=0 TO 200                                             541
518   FOR b=0 TO 320                                            515
519    farbe=POINT(b,a)                                         526
520    IF farbe>=0 THEN                                         163
521     PSET (b,a),pointer(farbe)                               041
522    END IF                                                   423
523   NEXT b                                                    712
524  NEXT a                                                     416
525  FOR a=0 TO 31                                              596
526   PALETTE
      a,Farben%(a,0)/16,Farben%(a,1)/16,Farben%(a,2)/
      16                                                        404
527  NEXT a                                                     780
528  GOTO warte                                                 984
529  zufall:                                                    568
530  RANDOMIZE TIMER                                            778
531  FOR a=0 TO 31                                              534
532   IF erlaubt(a)=0 THEN                                      322
533    FOR b=0 TO 2                                             221
534     Farben%(a,b)=INT(170*RND)/10                            203
535     IF Farben%(a,b)>16 THEN Farben%(a,b)=16                 232
536    NEXT b                                                   729
537    PALETTE
       a,Farben%(a,0)/16,Farben%(a,1)/16,Farben%(a,2)
       /16                                                      333
538   END IF                                                    147
539  NEXT a                                                     851
540  GOTO warte                                                 687
541  antik:                                                     664
542  FOR a=0 TO 31                                              852
543   IF erlaubt(a)=0 THEN                                      712
544    summe=Farben%(a,0)+Farben%(a,1)+Farben%(a,2)             312
545    summe=summe-3.2                                          155
546    IF summe<0 THEN summe=0                                  760
547    summe=summe/3                                            333
548    Farben%(a,0)=summe+1.6                                   030
549    Farben%(a,1)=summe+1.6                                   180
550    Farben%(a,2)=summe                                       411
551    FOR b=0 TO 2                                             894
552     IF Farben%(a,b)>16 THEN Farben%(a,b)=16                 495
553     IF Farben%(a,b)<0 THEN Farben%(a,b)=0                   892
554    NEXT b                                                   949
555    PALETTE
       a,Farben%(a,0)/16,Farben%(a,1)/16,Farben%(a,2)
       /16                                                      761
556   END IF                                                    503
557  NEXT a                                                     175
558  GOTO warte                                                 003
559  psng:                                                      929
560  FOR a=0 TO 31                                              593
561   IF erlaubt(a)=0 THEN                                      597
562    FOR b=0 TO 2                                             512
563     Farben%(a,b)=16-Farben%(a,b)                            963
564    NEXT b                                                   260
565    PALETTE
       a,Farben%(a,0)/16,Farben%(a,1)/16,Farben%(a,2)
       /16                                                      644
566   END IF                                                    850
567  NEXT a                                                     346
568  GOTO warte                                                 302
569  schwimmen:                                                 623
570  FOR a=31 TO 1 STEP -1                                      138
571   IF erlaubt(a)=0 THEN                                      502
572    FOR b=0 TO 2                                             337
573     summe=Farben%(a,b)-Farben%(a-1,b)                       790
574     summe=summe/8                                           224
575     Farben%(a,b)=Farben%(a,b)-summe                         407
576     Farben%(a-1,b)=Farben%(a-1,b)+summe                     716
577     IF Farben%(a,b)>16 THEN Farben%(a,b)=16                 181
578     IF Farben%(a,b)<0 THEN Farben%(a,b)=0                   082
579     IF Farben%(a-1,b)>16 THEN Farben%(a-1,b)=16             719
580     IF Farben%(a-1,b)<0 THEN Farben%(a-1,b)=0               456
```

```
   NEXT b                                              161
   PALETTE
    a,Farben%(a,0)/16,Farben%(a,1)/16,Farben%(a,2)
    /16                                                861
  END IF                                               571
 NEXT a                                                467
 GOTO warte                                            279
 kontr:                                                763
 FOR a=31 TO 1 STEP -1                                 614
  IF erlaubt(a)=0 THEN                                 658
   FOR b=0 TO 2                                        301
    summe=Farben%(a,b)-Farben%(a-1,b)                  810
    summe=summe/8                                      164
    Farben%(a-1,b)=Farben%(a-1,b)-summe                663
    Farben%(a,b)=Farben%(a,b)+summe                    560
    IF Farben%(a,b)>16 THEN Farben%(a,b)=16            297
    IF Farben%(a,b)<0 THEN Farben%(a,b)=0              142
    IF Farben%(a-1,b)>16 THEN Farben%(a-1,b)=16        315
    IF Farben%(a-1,b)<0 THEN Farben%(a-1,b)=0          708
   NEXT b                                              725
   PALETTE
    a,Farben%(a,0)/16,Farben%(a,1)/16,Farben%(a,2)
    /16                                                921
  END IF                                               383
 NEXT a                                                503
 GOTO warte                                            347
 fil:                                                  032
 bef=2:GOTO rgbquest                                   788
 fil2:                                                 311
 WINDOW CLOSE 5                                        590
 FOR a=0 TO 31                                         050
  IF erlaubt(a)=0 THEN                                 542
   Farben%(a,form)=0                                   288
   PALETTE
    a,Farben%(a,0)/16,Farben%(a,1)/16,Farben%(a,2)
    /16                                                624
  END IF                                               246
 NEXT a                                                246
 GOTO warte                                            658
 rgb:                                                  599
 bef=1                                                 670
 rgbquest:                                             785
 WINDOW 5,"RGB-Töne",(10,60)-(310,140),0,1             403
 COLOR 0,1                                             779
 CLS:LOCATE 2,2:PRINT "Bitte geben Sie ein, für"       248
 PRINT " welche Farbe ihr Befehl gilt!"                071
 SOUND 880,10,100                                      286
 LOCATE 5,5:PRINT "Rot";                               555
 LOCATE 5,13:PRINT "Grün";                             853
 LOCATE 5,21:PRINT "Blau";                             187
 LOCATE 5,29:PRINT "OK";                               671
 LINE (30,30)-(74,42),0,b                              171
 LINE (94,30)-(130,42),0,b                             870
 LINE (158,30)-(194,42),0,b                            748
 LINE (222,30)-(258,42),0,b                            879
 warte10:                                              605
 Test=MOUSE(0)                                         760
 WHILE MOUSE(0)<>0                                     113
  x=MOUSE(1):y=MOUSE(2)                                577
  IF y>30 AND y<42 THEN                                959
   IF x>38 AND x<74 THEN form=0:ON bef GOTO
    rgb2,fil2                                          652
   IF x>94 AND x<130 THEN form=1:ON bef GOTO
    rgb2,fil2                                          561
   IF x>158 AND x<194 THEN form=2:ON bef GOTO
    rgb2,fil2                                          358
   IF x>222 AND x<258 THEN WINDOW CLOSE 5:GOTO
    warte                                              087
  END IF                                               717
 WEND                                                  415
 GOTO warte10                                          744
 rgb2:                                                 091
 WINDOW CLOSE 5                                        978
 FOR a=0 TO 31                                         934
  IF erlaubt(a)=0 THEN                                 954
   summe=Farben%(a,0)+Farben%(a,1)+Farben%(a,2)        906
   IF summe<=16 THEN                                   963
    Farben%(a,0)=0:Farben%(a,1)=0:Farben%(a,2)=0:F
    arben%(a,form)=summe                               143
   ELSE                                                208
    Farben%(a,form)=16                                 741
    summe=summe-16                                     864
    summe=summe/2                                      020
    FOR b=0 TO 2                                       527
     IF form<>b THEN Farben%(a,b)=summe                832
    NEXT b                                             089
   END IF                                              063
   PALETTE
    a,Farben%(a,0)/16,Farben%(a,1)/16,Farben%(a,2)
    /16                                                931
  END IF                                               209
 NEXT a                                                057
 GOTO warte                                            325
 grau:                                                 014
 FOR a=0 TO 31                                         778
  IF erlaubt(a)=0 THEN                                 094
   summe=Farben%(a,0)+Farben%(a,1)+Farben%(a,2)        726
   summe=summe/3                                       411
   Farben%(a,0)=summe:Farben%(a,1)=summe:Farben%(a
   ,2)=summe                                           213
   PALETTE
    a,Farben%(a,0)/16,Farben%(a,1)/16,Farben%(a,2)
    /16                                                409
  END IF                                               919
 NEXT a                                                311
 GOTO warte                                            667
 hd:                                                   581
 WINDOW 5,"Heller - Dunkler",(30,20)-(260,120),0,1     832
 LOCATE 4,3:PRINT "Heller";                            101
 LOCATE 4,13:PRINT "Dunkler";                          780
 LOCATE 4,23:PRINT "OK";                               705
 LINE (14,22)-(72,34),1,b                              152
 LINE (94,22)-(152,34),1,b                             962
 LINE (174,22)-(208,34),1,b                            061
 warte8:                                               034
 Test=MOUSE(0)                                         961
 WHILE MOUSE(0)<>0                                     998
 x=MOUSE(1):y=MOUSE(2)                                 514
 IF y>22 AND y<34 THEN                                 994
  IF x>14 AND x<72 THEN                                171
   FOR a=0 TO 31                                       795
    IF erlaubt(a)=0 THEN                               343
     FOR b=0 TO 2                                      122
      IF Farben%(a,b)/16+.05 < 1 THEN                  928
       Farben%(a,b)=Farben%(a,b)+(.05*16)              163
      END IF                                           441
     NEXT b                                            970
     PALETTE
      a,Farben%(a,0)/16,Farben%(a,1)/16,Farben%(a,
      2)/16                                            722
    END IF                                             968
   NEXT a                                              672
  ELSEIF x>94 AND x<152 THEN                           329
   FOR a=0 TO 31                                       345
    IF erlaubt(a)=0 THEN                               013
     FOR b=0 TO 2                                      592
      IF Farben%(a,b)/16-.05 >0 THEN                   861
       Farben%(a,b)=Farben%(a,b)-(.05*16)              786
      END IF                                           520
     NEXT b                                            761
     PALETTE
      a,Farben%(a,0)/16,Farben%(a,1)/16,Farben%(a,
      2)/16                                            725
    END IF                                             987
   NEXT a                                              603
  ELSEIF x>174 AND x<208 THEN                          713
   WINDOW CLOSE 5:GOTO warte                           838
  END IF                                               900
 END IF                                                106
 WEND                                                  652
 GOTO warte8                                           968
 projekt:                                              857
 ON b GOTO
  laden,speichern,Farben,ende,drucken,vorein           163
 vorein:                                               356
 WINDOW 5,"Preferences",(0,20)-(310,130),0,1           157
 FOR a=0 TO 31                                         861
  LINE (a*8,1)-(a*8+8,13),a,bf                         703
  LOCATE 2,a+1                                         370
  IF erlaubt(a)=0 THEN PRINT "J";: ELSE PRINT "N";     501
 NEXT a                                                829
 LOCATE 4,8:PRINT "+[2SPACES]Mosaik -[2SPACES]X:";     377
 PRINT USING "##";mosaikx;                             344
 PRINT " -";                                           409
 LOCATE 6,8:PRINT "+[2SPACES]Mosaik -[2SPACES]Y:";     156
 PRINT USING "##";mosaiky;                             180
 PRINT " -";                                           933
 LINE (195,65)-(225,85),1,b                            231
 LOCATE 10,26:PRINT "OK";                              716
 vorwarte:                                             460
 Test=MOUSE(0)                                         707
 WHILE MOUSE(0)=0:WEND                                 415
 x=MOUSE(1):y=MOUSE(2)                                 375
 IF y>1 AND y<13 AND x<256 THEN                        389
  farbe=INT (x/8)                                      945
  IF erlaubt(farbe)=0 THEN                             152
   erlaubt(farbe)=1:LOCATE 2,farbe+1:PRINT "N";        550
  ELSE                                                 087
   erlaubt(farbe)=0:LOCATE 2,farbe+1:PRINT "J";        824
  END IF                                               382
  FOR a=1 TO 300:NEXT a                                164
 ELSEIF x>50 AND x<70 AND y>20 AND y<38 THEN           623
  IF mosaikx<99 THEN mosaikx=mosaikx+1                 944
  LOCATE 4,23:PRINT USING"##";mosaikx;                 054
  FOR a=1 TO 100:NEXT a                                010
 ELSEIF x>200 AND x<220 AND y>20 AND y<38 THEN         703
  IF mosaikx>1 THEN mosaikx=mosaikx-1                  115
  LOCATE 4,23:PRINT USING"##";mosaikx;                 297
  FOR a=1 TO 100:NEXT a                                041
 ELSEIF x>50 AND x<70 AND y>38 AND y<50 THEN           463
  IF mosaiky<99 THEN mosaiky=mosaiky+1                 907
  LOCATE 6,23:PRINT USING"##";mosaiky;                 432
  FOR a=1 TO 100:NEXT a                                476
 ELSEIF x>200 AND x<220 AND y>38 AND y<50 THEN         928
  IF mosaiky>1 THEN mosaiky=mosaiky-1                  303
  LOCATE 6,23:PRINT USING"##";mosaiky;                 404
  FOR a=1 TO 100:NEXT a                                200
 ELSEIF x>195 AND x<225 AND y>65 AND y<85 THEN         690
  WINDOW CLOSE 5:GOTO warte                            531
 END IF                                                937
 GOTO vorwarte                                         412
 drucken:                                              866
REM *** Hardcopyroutine by Carolyn Scheppner  CBM
 ***                                                   866
BorderFlag% = 0                                        060
sWindow&    = WINDOW(7)                                602
sScreen&    = PEEKL(sWindow& + 46)                     524
sViewPort& = sScreen& + 44                             522
sRastPort& = sScreen& + 84                             071
sColorMap& = PEEKL(sViewPort& + 4)                     436
maxWidth%  = PEEKW(sScreen& + 12)                      116
maxHeight% = PEEKW(sScreen& + 14)                      015
viewModes% = PEEKW(sViewPort& + 32)                    142
command%   = 11                                        525
srcX% = 0                                              987
srcY% = 0                                              086
srcWidth%  = maxWidth%                                 415
srcHeight% = maxHeight%                                738
destRows& = 0                                          908
destCols& = 0                                          428
special% = &H84                                        889
IF BorderFlag% = 0 THEN                                585
```

```
   srcX% = srcX% + 3                                    050
   srcY% = srcY% + 11                                   268
   srcWidth%  = srcWidth% - 3 - 11                      350
   srcHeight% = srcHeight% - 11 - 3                     206
END IF                                                  716
sigBit% =  AllocSignal%(-1)                             793
ClearPublic& = 65537&                                   730
msgPort& = AllocMem&(40,ClearPublic&)                   394
IF msgPort& = 0 THEN                                    063
   PRINT "msgPort nicht allokierbar."                   488
   GOTO dcleanup4                                       861
END IF                                                  571
POKE(msgPort& + 8), 4                                   196
POKE(msgPort& + 9), 0                                   830
portName$ = "MyPrtPort"+CHR$(0)                         671
POKEL(msgPort& + 10), SADD(portName$)                   280
POKE(msgPort& + 14), 0                                  442
POKE(msgPort& + 15), sigBit%                            864
sigTask& = FindTask&(0)                                 084
POKEL(msgPort& + 16), sigTask&                          028
CALL AddPort(msgPort&)                                  529
ioRequest& = AllocMem&(64,ClearPublic&)                 140
IF ioRequest& = 0  THEN                                 122
   PRINT "ioRequest nicht allokierbar."                 236
   GOTO dcleanup3                                       448
END IF                                                  454
POKE(ioRequest& + 8),5                                  977
POKE(ioRequest& + 9),0                                  276
POKEL(ioRequest& + 14), msgPort&                        249
devName$ = "printer.device"+CHR$(0)                     353
pError& =
  OpenDevice&(SADD(devName$),0,ioRequest&,0)            411
IF pError& <> 0  THEN                                   671
   PRINT "Drucker nicht ansprechbar."                   229
   GOTO dcleanup2                                       208
END IF                                                  062
POKEW(ioRequest& + 28), command%                        449
POKEL(ioRequest& + 32), sRastPort&                      396
POKEL(ioRequest& + 36), sColorMap&                      221
POKEL(ioRequest& + 40), viewModes%                      695
POKEW(ioRequest& + 44), srcX%                           365
POKEW(ioRequest& + 46), srcY%                           282
POKEW(ioRequest& + 48), srcWidth%                       460
POKEW(ioRequest& + 50), srcHeight%                      032
POKEL(ioRequest& + 52), destCols&                       712
POKEL(ioRequest& + 56), destRows&                       214
POKEW(ioRequest& + 60), special%                        639
ioError& = DoIO&(ioRequest&)                            141
IF ioError& <> 0 THEN                                   221
   PRINT "DumpRPort Fehler =" ioError&                  278
   GOTO dcleanup1                                       832
END IF                                                  814
dcleanup1:                                              369
   CALL CloseDevice(ioRequest&)                         023
dcleanup2:                                              367
   POKE(ioRequest& + 8), &HFF                           999
   POKEL(ioRequest& + 20), -1                           997
   POKEL(ioRequest& + 24), -1                           351
   CALL FreeMem(ioRequest&,64)                          146
dcleanup3:                                              271
   CALL RemPort(msgPort&)                               843
   POKE(msgPort& + 8), &HFF                             610
   POKEL(msgPort& + 20), -1                             123
   CALL FreeSignal(sigBit%)                             795
   CALL FreeMem(msgPort&,40)                            667
dcleanup4:                                              789
   GOTO warte                                           025
  ende:                                                 711
  WINDOW 5,"Programm beenden",(50,60)-(250,140),0,1     857
  COLOR 0,1                                             985
  CLS                                                   463
  LOCATE 2,2:PRINT "Wollen Sie mir das"                 076
  PRINT "[2SPACES]wirklich antun ?"                     087
  LOCATE 5,5:PRINT "JA";                                290
  LOCATE 5,10:PRINT "NEIN";                             820
  LINE (30,30)-(67,42),0,b                              982
  LINE (70,30)-(107,42),0,b                             783
  SOUND 880,10,100                                      510
  warte7:                                               058
   WHILE MOUSE(0)>-1:WEND                               449
   x=MOUSE(1):y=MOUSE(2)                                081
   IF y>30 AND y<42 THEN                                911
    IF x>30 AND x<67 THEN LIBRARY CLOSE:MENU
     RESET:WINDOW CLOSE 5:WINDOW                        070
CLOSE 3:SCREEN CLOSE 1:SYSTEM                           415
    IF x>70 AND x<107 THEN WINDOW CLOSE 5:GOTO
     warte                                              926
   END IF                                               796
  GOTO warte7                                           407
speichern:                                              834
  GOSUB filename                                        495
  IF ILBMNam$="" THEN warte                             365
  f$ = ILBMNam$                                         432
REM *** IFF - Laderoutine by Carolyn Scheppner  CBM
  ***                                                   432
 fHandle& = 0                                           917
 mybuf& = 0                                             959
 filename$ = f$ + CHR$(0)                               424
 fHandle& = xOpen&(SADD(filename$),1006)                439
IF fHandle& = 0 THEN                                    334
   saveError$ = "Ausgabedatei nicht erzeugbar."         558
   GOTO Scleanup2                                       640
END IF                                                  110
ClearPublic& = 65537&                                   251
mybufsize& = 120                                        907
mybuf& = AllocMem&(mybufsize&,ClearPublic&)             862
IF mybuf& = 0 THEN                                      726
   saveError$ = "Pufferspeicher nicht verfügbar."       778
   GOTO Scleanup2                                       252
END IF                                                  010
cbuf& = mybuf&                                          310
GOSUB GetScrAddrs                                       348
zero& = 0                                               687
pad%  = 0                                               470
aspect% = &HA0B                                         961
BMHDsize& = 20                                          028
CMAPsize& = (2^scrDepth%) * 3                           611
CAMGsize& = 4                                           897
CCRTsize& = 14                                          904
BODYsize& = (scrWidth%/8) * scrHeight% * scrDepth%      797
FORMsize& =
 BMHDsize&+CMAPsize&+CAMGsize&+CCRTsize&+BODYsize&+
 44                                                     497
tt$ = "FORM"                                            134
wLen& = xWrite&(fHandle&,SADD(tt$),4)                   754
wLen& = xWrite&(fHandle&,VARPTR(FORMsize&),4)           966
tt$ = "ILBM"                                            071
wLen& = xWrite&(fHandle&,SADD(tt$),4)                   119
IF wLen& <= 0 THEN                                      878
   saveError$ = "Schreibfehler beim FORM-Header."       999
   GOTO Scleanup2                                       313
END IF                                                  543
tt$ = "BMHD"                                            007
wLen& = xWrite&(fHandle&,SADD(tt$),4)                   311
wLen& = xWrite&(fHandle&,VARPTR(BMHDsize&),4)           346
wLen& = xWrite&(fHandle&,VARPTR(scrWidth%),2)           900
wLen& = xWrite&(fHandle&,VARPTR(scrHeight%),2)          911
wLen& = xWrite&(fHandle&,VARPTR(zero&),4)               816
temp% = (256 * scrDepth%)                               634
wLen& = xWrite&(fHandle&,VARPTR(temp%),2)               718
wLen& = xWrite&(fHandle&,VARPTR(zero&),4)               323
wLen& = xWrite&(fHandle&,VARPTR(aspect%),2)             121
wLen& = xWrite&(fHandle&,VARPTR(scrWidth%),2)           423
wLen& = xWrite&(fHandle&,VARPTR(scrHeight%),2)          954
IF wLen& <= 0 THEN                                      385
   saveError$ = "Schreibfehler beim BMHD-Chunk."        165
   GOTO Scleanup2                                       095
END IF                                                  149
tt$ = "CMAP"                                            251
wLen& = xWrite&(fHandle&,SADD(tt$),4)                   987
wLen& = xWrite&(fHandle&,VARPTR(CMAPsize&),4)           188
FOR kk = 0 TO nColors% - 1                              192
   regTemp% = PEEKW(colorTab& + (2*kk))                 549
   POKE(cbuf&+(kk*3)),(regTemp% AND &HF00) / 16         073
   POKE(cbuf&+(kk*3)+1),(regTemp% AND &HF0)             915
   POKE(cbuf&+(kk*3)+2),(regTemp% AND &HF) * 16         275
NEXT                                                    130
wLen& = xWrite&(fHandle&,cbuf&,CMAPsize&)               925
IF wLen& <= 0 THEN                                      284
   saveError$ = "Schreibfehler beim CMAP-Chunk."        114
   GOTO Scleanup2                                       524
END IF                                                  178
tt$ = "CAMG"                                            643
wLen& = xWrite&(fHandle&,SADD(tt$),4)                   603
wLen& = xWrite&(fHandle&,VARPTR(CAMGsize&),4)           255
vpModes& = PEEKW(sViewPort& + 32)                       166
wLen& = xWrite&(fHandle&,VARPTR(vpModes&),4)            181
IF wLen& <= 0 THEN                                      012
   saveError$ = "Schreibfehler beim CAMG-Chunk"         503
   GOTO Scleanup2                                       225
END IF                                                  647
tt$ = "CCRT"                                            232
wLen& = xWrite&(fHandle&,SADD(tt$),4)                   884
wLen& = xWrite&(fHandle&,VARPTR(CCRTsize&),4)           876
wLen& = xWrite&(fHandle&,VARPTR(ccrtDir%),2)            881
temp% = (256*ccrtStart%) + ccrtEnd%                     271
wLen& = xWrite&(fHandle&,VARPTR(temp%),2)               767
wLen& = xWrite&(fHandle&,VARPTR(ccrtSecs&),4)           630
wLen& = xWrite&(fHandle&,VARPTR(ccrtMics&),4)           095
wLen& = xWrite&(fHandle&,VARPTR(pad%),2)                250
IF wLen& <= 0 THEN                                      193
   saveError$ = "Schreibfehler beim CCRT-Chunk."        810
   GOTO Scleanup2                                       004
END IF                                                  234
tt$ = "BODY"                                            225
wLen& = xWrite&(fHandle&,SADD(tt$),4)                   385
wLen& = xWrite&(fHandle&,VARPTR(BODYsize&),4)           987
scrRowBytes% = scrWidth% / 8                            052
FOR rr = 0 TO scrHeight% -1                             339
   FOR pp = 0 TO scrDepth% -1                           782
      scrRow& = bPlane&(pp)+(rr*scrRowBytes%)           032
      wLen& =
       xWrite&(fHandle&,scrRow&,scrRowBytes%)           389
      IF wLen& <= 0 THEN                                324
         saveError$ = "Schreibfehler beim
           BODY-Chunk."                                 647
         GOTO Scleanup2                                 753
      END IF                                            023
   NEXT                                                 478
NEXT                                                    125
saveError$ = ""                                         271
Scleanup2:                                              914
IF fHandle& <> 0 THEN CALL xClose&(fHandle&)            815
IF mybuf& <> 0 THEN CALL
 FreeMem&(mybuf&,mybufsize&)                            013
CLOSE                                                   675
GOTO warte                                              127
  Farben:                                               799
  fault%=0                                              573
  WINDOW 5,"Farbenmenu",(4,20)-(280,70),0,1             120
  COLOR 1,0:CLS                                         864
  FOR a=0 TO 31                                         252
   LINE (a*4,3)-(a*4+3,10),a,bf                         596
  NEXT a                                                484
  LINE (210,3)-(220,10),zeifarbe,bf                     158
  LINE (10,12)-(110,20),1,b                             428
  LINE (10,22)-(110,30),1,b                             632
  LINE (10,32)-(110,40),1,b                             530
  LINE (190,35)-(245,50),1,b                            394
  FOR a=1 TO 4:LINE
   (160+(a-1)*25+10,15)-(160+a*25+9,35),1,b:NEXT a      870
  LOCATE 6,26:PRINT "OK";                               480
  LOCATE 4,23:PRINT "Mi";:LOCATE 4,26:PRINT
   "Sp";:LOCATE 4,29:PRINT                              716
```

```
"Co";:LOCATE 4,32:PRINT "Ch";                              701
  LOCATE 3,16:PRINT "RGB";                                 272
  frgb:                                                    547
  LINE (120,24)-(127,32),gesetzt%(0) XOR 1,bf              792
  LINE (128,24)-(135,32),gesetzt%(1) XOR 1,bf              657
  LINE (136,24)-(143,32),gesetzt%(2) XOR 1,bf              217
  r=Farben%(zeifarbe,0)/16*100                             721
  g=Farben%(zeifarbe,1)/16*100                             123
  b=Farben%(zeifarbe,2)/16*100                             349
  LINE (11,13)-(109,19),0,bf                               410
  LINE (11,23)-(109,29),0,bf                               869
  LINE (11,33)-(109,39),0,bf                               814
  LINE (r+10,13)-(r+10,19),1                               280
  LINE (g+10,23)-(g+10,29),1                               318
  LINE (b+10,33)-(b+10,39),1                               604
  warte5:                                                  846
  WINDOW OUTPUT 5                                          396
  Test=MOUSE(0)                                            899
  WHILE MOUSE(0)=0:WEND                                    887
  x=MOUSE(1):y=MOUSE(2)                                    751
  IF y>12 AND y<45 AND x>5 AND x<120 THEN                  620
   IF x>=10 AND x<=110 THEN                                076
    IF y>12 AND y<20 THEN                                  250
     r=x-10                                                475
    ELSEIF y>22 AND y<30 THEN                              032
     g=x-10                                                618
    ELSEIF y>32 AND y<40 THEN                              809
     b=x-10                                                514
    END IF                                                 352
   END IF                                                  078
   Farben%(zeifarbe,0)=r/100*16                            466
   Farben%(zeifarbe,1)=g/100*16                            880
   Farben%(zeifarbe,2)=b/100*16                            046
   PALETTE                                                 709
zeifarbe,Farben%(zeifarbe,0)/16,Farben%(zeifarbe,1)
/16,Farben%(zeifarbe,2)                                    980
/16                                                        054
   GOTO frgb                                               404
  ELSEIF y>35 AND y<50 AND x>190 AND x<245 THEN            916
   WINDOW CLOSE 5:GOTO warte                               845
  ELSEIF x>170 AND x<270 AND y>15 AND y<35 THEN            878
   GOSUB request                                           035
   IF fault%=1 THEN                                        412
    f$="Farbenmenu":CALL
     SetWindowTitles(WINDOW(7),SADD(f$+CHR$(0)),0)         670
    fault%=0:GOTO frgb                                     988
   ELSEIF x>170 AND x<195 THEN                             199
    FOR a=0 TO 2                                           365
     IF gesetzt%(a)=0 THEN                                 753
      summe=Farben%(farbe1,a)+Farben%(farbe2,a)            693
      summe=summe/2                                        021
      Farben%(farbe1,a)=summe                              435
      Farben%(farbe2,a)=summe                              542
     END IF                                                436
    NEXT a                                                 556
   ELSEIF x>195 AND x<220 AND farbe1<>farbe2 THEN          251
    ox=1:IF farbe2<farbe1 THEN ox=-1                       381
    FOR a=0 TO 2                                           347
     IF gesetzt%(a)=0 THEN                                 511
      c=Farben%(farbe1,a)                                  721
      summe=Farben%(farbe2,a)-Farben%(farbe1,a)            183
      summe=summe/(farbe2-farbe1)                          702
      FOR b=0 TO (farbe2-farbe1) STEP ox                   759
       Farben%(farbe1+b,a)=c+b*summe                       400
      NEXT b                                               041
     END IF                                                423
    NEXT a                                                 711
    FOR a=farbe1 TO farbe2 STEP ox                         609
     PALETTE
      a,Farben%(a,0)/16,Farben%(a,1)/16,Farben%(a,2
     )/16                                                  541
    NEXT a                                                 037
   ELSEIF x>220 AND x<245 THEN                             500
    FOR a=0 TO 2                                           674
     IF gesetzt%(a)=0 THEN                                 370
      Farben%(farbe2,a)=Farben%(farbe1,a)                  300
     END IF                                                698
    NEXT a                                                 898
   ELSEIF x>245 AND x<270 THEN                             082
    FOR a=0 TO 2                                           904
     IF gesetzt%(a)=0 THEN                                 152
      SWAP Farben%(farbe1,a),Farben%(farbe2,a)             240
     END IF                                                886
    NEXT a                                                 862
   END IF                                                  244
   PALETTE                                                 739
farbe1,Farben%(farbe1,0)/16,Farben%(farbe1,1)/16,Fa
rben%(farbe1,2)/16                                         872
   PALETTE                                                 703
farbe2,Farben%(farbe2,0)/16,Farben%(farbe2,1)/16,Fa
rben%(farbe2,2)/16                                         544
   GOTO frgb                                               718
  ELSEIF x>1 AND x<127 AND y>=3 AND y=<10 THEN             865
   zeifarbe=POINT(x,y):LINE
    (210,3)-(220,10),zeifarbe,bf                           310
   GOTO frgb                                               684
  ELSEIF x>120 AND x<144 AND y>24 AND y<32 THEN            669
   IF x>120 AND x<128 THEN gesetzt%(0)=gesetzt%(0)
    XOR 1                                                  105
   IF x>128 AND x<136 THEN gesetzt%(1)=gesetzt%(1)
    XOR 1                                                  886
   IF x>136 AND x<144 THEN gesetzt%(2)=gesetzt%(2)
    XOR 1                                                  135
   FOR a=1 TO 100:NEXT a                                   719
   GOTO frgb                                               017
  ELSEIF WINDOW (0)=2 THEN                                 188
   WINDOW OUTPUT 2                                         199
   x=MOUSE(1):y=MOUSE(2)                                   279
   IF POINT(x,y)>=0 THEN                                   560
    zeifarbe=POINT(x,y)                                    573
    WINDOW OUTPUT 5                                        055
    LINE (210,3)-(220,10),zeifarbe,bf                      573
     GOTO frgb                                             871
    END IF                                                 349
    WINDOW OUTPUT 5                                        983
   END IF                                                  589
   GOTO warte5                                             442
   request:                                                021
    FOR a=1 TO 350:NEXT a                                  716
    f$="Geben Sie die 1.Farbe an !!"                       839
    CALL
     SetWindowTitles(WINDOW(7),SADD(f$+CHR$(0)),0)         381
    Test=MOUSE(0):WHILE MOUSE(0)=0:WEND                    906
    ox=MOUSE(1):oy=MOUSE(2)                                628
    IF ox>1 AND ox<127 AND oy>=3 AND oy<=10 THEN
     farbe1=POINT(ox,oy) ELSE                              379
fault%=1:RETURN                                            900
    FOR a=1 TO 250:NEXT a                                  842
    f$="Geben Sie jetzt die 2.Farbe an !"                  934
    CALL
     SetWindowTitles(WINDOW(7),SADD(f$+CHR$(0)),0)         348
    Test=MOUSE(0):WHILE MOUSE(0)=0:WEND                    445
    ox=MOUSE(1):oy=MOUSE(2)                                315
    IF ox>1 AND ox<127 AND oy>=3 AND oy<=10 THEN
     farbe2=POINT(ox,oy) ELSE                              967
fault%=1:RETURN                                            792
    f$="Farbenmenu"                                        986
    CALL
     SetWindowTitles(WINDOW(7),SADD(f$+CHR$(0)),0)         688
    RETURN                                                 648
 filename:                                                 611
  WINDOW 5,"Bitte Filenamen
   eingeben",(0,0)-(311,8),0,1                             469
  COLOR 1,0                                                589
  CLS                                                      851
  LINE INPUT ILBMNam$                                      063
  WINDOW CLOSE 5                                           022
  RETURN                                                   342
  laden:                                                   772
  GOSUB filename                                           889
  IF ILBMNam$="" THEN warte                                535
  GOTO neuladen                                            444
  endeladen:                                               602
  CLOSE:GOTO warte                                         074
```

Farbhokuspokus ist mit Zaubermeister kein Problem.

AmigaWelt-Autoren-Coupon

Programmeinsendungen ist dieser Coupon ausgefüllt beizulegen. Alle Beiträge nehmen am Wettbewerb „Superlisting“ teil.

Name: ______________________

Vorname: ______________________

Straße: ______________________

Ort: ______________________

Land: ______________________

Telefon: ______________________

Geburtsdatum: ______________________

Programmname: ______________________

Die Sendung muß enthalten:

1. Programmbeschreibung auf Diskette als ASCII-File (unbedingt)
2. Programm auf Diskette (wichtig: Basic-Programme als asc-File)

☐ Sonstiges: ______________________

Bitte alle Teile der Einsendung mit Name, Adresse und Tel.-Nr. versehen!

Ich bin damit einverstanden, daß die Rechte an meinem Programm an den Verlag (CW-Publikationen) übergehen, sobald der Verlag die Veröffentlichung zugesagt hat. Die Honorierung erfolgt nach Veröffentlichung entsprechend den verlagsüblichen Sätzen. Weiterhin bestätige ich mit meiner Unterschrift, daß mein Programm frei von Rechten Dritter ist und auch keinem anderen Verlag oder einer anderen Firma als Angebot vorliegt.

Unterschrift: ______________________

Bei Minderjährigen (unter 18 Jahren) bitte die Unterschrift eines Erziehungsberechtigten:

Unterschrift: ______________________

Nigel Mansell's Grand Prix

Mit 300 aus der Kurve

In Formel-I-Rennen ist es lebenswichtig, Kurventechniken zu beherrschen, Eigenarten der Rennstrecken zu kennen und mit siegeshungrigen Gegnern fertig zu werden. Eine Kostprobe davon bietet „Nigel Mansell's Grand Prix". Hier müssen sich Anfänger erst einmal daran gewöhnen, aus nahezu jeder Kurve zu fliegen und den größten Teil des Rennens auf dem Seitenstreifen zu verbringen. Fahrer, die die Sachskurve des Hockenheimrings im sechsten Gang mit 300 km/h nehmen, haben den Purzelbaum in die Pampa verdient.

Vor dem ersten Tritt aufs Gaspedal wird ein Trainingsparcours ausgewählt. 16 weltbekannte Formel-I-Rennstrecken stehen zur Wahl — darunter Monza, Monaco, Silverstone und der erst seit zwei Jahren von Formel-I-Wagen befahrene Hungaro-Ring. Nach dem Blick auf die Strecken, ein Blick ins Innere des Rennwagens. Auch das Cockpit ist auf realistisch getrimmt. Man findet Digitalanzeigen für Verbrauch, Geschwindigkeit, durchschnittliches Tempo und beste Rundenzeit, ferner den Tourenzähler und Balkenanzeigen für Öl- und Wassertemperatur. Es gibt sogar eine Einrichtung, die aus Verbrauch und Spritreserve die Reichweite des Wagens errechnet.

Die beste Startposition bekommt derjenige Fahrer, der während des Trainings die beste Zeit gefahren ist. So ein Vorsprung muß verteidigt werden. Das wird dadurch erschwert, daß die computergesteuerten Wagen, falls man ihnen nicht schnell ausweicht, erbarmungslos auffahren. Der Wagen gerät ins Schleudern. Das ist zwar nicht gerade wirklichkeitsnah, doch zwingt es dazu, den Rückspiegel zu benutzen.

Je nach Ausdauer kann man Rennen mit zwei, fünf, zehn oder 25 Runden fahren. Bei den längeren Strecken müssen Boxenstops zum Tanken und Reifenwechseln eingelegt werden. Die Darstellung dieser Stops ist allerdings recht dürftig ausgefallen — es gibt keine Grafik und schon gar keine Animation. Es erscheint lediglich die Meldung, daß der Wagen wieder okay ist.

(A. Albert)



Mit 1000 Pferdestärken gegen den Horizont.

Preis: 80 Mark	
Hersteller: Martech	
Vertrieb: Rushware	
Spaß	7
Grafik	6
Sound	7
Idee	5
Schwierigkeit	8

---

4 × 4 Off Road Racing

Ralley ohne Regeln

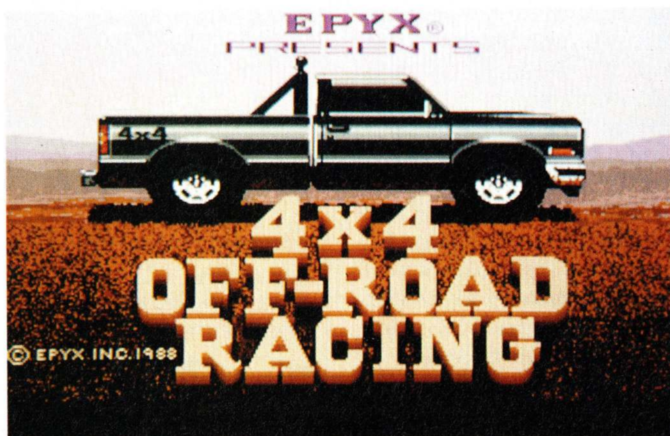
In Epyx' Ralley-Simulation werden Höchstleistungen von Mensch und Maschine verlangt. Es geht durch den glühend heißen Sand des Death Valley, weiter durch die schmatzenden Sümpfe Georgias und über den felsigen Untergrund der Baja California und durch den dick verschneiten Staat Michigan.

Für jede Tour wählt der Fahrer das geeignete Fahrzeug. Je nach Bodenbeschaffenheit müssen Reifen des Typs Normal, Mudder oder All-Terrain aufgezogen werden. Im Custom Shop kann ein Verdeck zur Vergrößerung des Laderaums besorgt werden. Ferner sollte für alle Fälle an Seilwinde, Ersatzreifen und Reservetank gedacht werden. Zuletzt werden Werkzeug und Ersatzteile eingepackt. Im Automarkt – kaum zu glauben! – steht sogar ein Mechaniker im Regal.

Endlich starten die tollkühnen Männer mit ihren vollgepackten Kisten. Auf den ersten paar Meilen macht sich der Fahrer mit der eigenwilligen Steuerung vertraut (wird der Joystick nach vorn gedrückt, bremst das Auto). Plötzlich sieht er Zweige und Steine auf der Piste. Wer nicht geistesgegenwärtig ausweicht, erlebt die interessante, aber unrealistische Reaktion seines Autos: sobald es ein kleines Hindernis berührt, legt es sich auf die Seite und fährt auf zwei Rädern weiter. Ein Zusammenstoß mit großen Felsbrocken hat einen heftigen Überschlag zur Folge. Allerdings leidet das Gefährt unter häufigen Saltos. Da fängt plötzlich der Motor an zu kochen, das Getriebe ist kaputt, und wertvolle Minuten werden mit Reparaturen vergeudet. Bei einer Panne wird offensichtlich, wer Anfänger und wer Profi ist. Während das Greenhorn mit Hammerschlägen reparieren muß, hat ein Profi alles dabei und bringt den Schaden schnell in Ordnung. Ein alter Rallye-Hase wird auf der Tourenkarte auch immer nach Abkürzungen suchen, so daß der brav dahintuckernde Anfänger bestenfalls dessen Staubwolke am Horizont sieht.

Der Gewinner aller vier Strecken, von der zweiten Schwierigkeitsstufe aufwärts, darf sich King of the Road nennen. Und das hat er sich redlich verdient — die Parcours sind hart und die Gegner unerbittlich.

(A. Albert)



Auf steinigen Pisten und zugefrorenen Flüssen beweist der Geländewagen seine Fähigkeit zu Höchstleistungen.

Preis: 60 Mark	
Hersteller: Epyx	
Vertrieb: Leisuresoft	
Spaß	8
Grafik	7
Sound	6
Idee	8
Schwierigkeit	7

Fish

Ein Goldfisch ermittelt

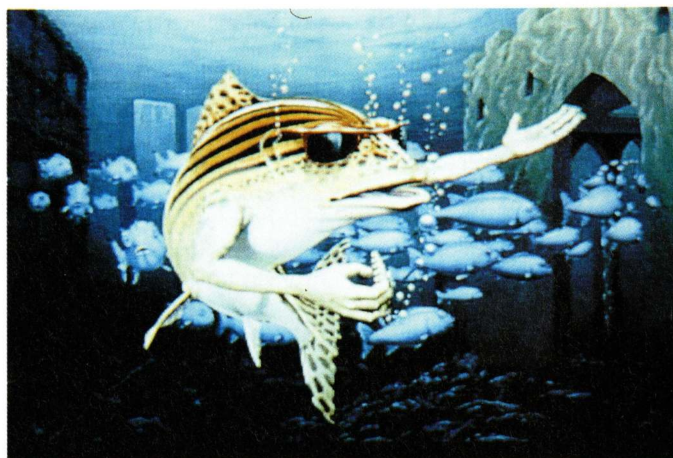
Verrückter geht's nicht! Im Adventure „Fish!“ wechselt ein Agent namens Blowtorch den Körper — so selbstverständlich wie normale Menschen ihre Socken wechseln. Micky Blowtorch hat interdimensionale Fähigkeiten. Er kann sich dank seiner mentalen Kräfte in die Körper anderer Personen oder Wesen versetzen. Dadurch fühlt, hört und sieht er genau das, was auch der Wirtskörper wahrnimmt. Und dies, obwohl sich Mickys eigener Körper an einem anderen Ort befindet. Der Wechsel von einem Individuum zum anderen ist jedoch kein Zuckerlecken. „Es ist wie bei einem Autounfall, wenn man selber das Auto ist“, sagt Micky Blowtorch, „es tut unheimlich weh.“ Allerdings nimmt Micky den Schmerz kaum noch wahr, da er von einem Auftrag zum anderen hetzt und den häufigen Körperwechsel bereits gewöhnt ist.

Bei einem solchen Job braucht Micky hin und wieder einen Erholungsurlaub. Dazu versetzt er sich in den Körper eines Goldfisches. Urlaubsort ist ein Goldfischglas. Aber nicht einmal als Goldfisch hat man seine Ruhe! Irgendjemand wirft nämlich ein eigenartiges Spielzeugschlößchen in den nassen Urlaubsort. Das bedeutet einen neuen Auftrag! Mit Urlaubsruhe und Frieden ist es vorbei. Was passiert ist! Seit einiger Zeit treiben die Seven Deadly Fins (die sieben tödlichen Flossen) ihr Unwesen auf einem von Wasser bedeckten Planeten. Dessen Bewohner sind halb Fisch, halb Mensch — „Finschen“ sozusagen. Aus ungeklärten Gründen verdampft das Wasser langsam, aber stetig, so daß die Finschen früher oder später auf dem Trockenen sitzen werden. Da das Wasser ihr Lebenselement ist und sie durch Kiemen atmen, bekäme ihnen das sicherlich schlecht. Hinter dem rätselhaften Wasserschwund stecken natürlich die Deadly Fins.

Betrachtet Micky das Spielzeugschlößchen genauer, entdeckt er drei Dimensionstore. Durch diese wunderlichen Tore muß er zu drei verschiedenen Orten schwimmen, um drei einzigartige Artefakte zu finden. Mit Hilfe dieser Gegenstände können die Deadly Fins besiegt und die Finschen vor dem Verdorren gerettet werden. Das erste Dimensionstor führt mitten in den tropischen Regenwald. Micky versucht dort einen Freak davon abzubringen, Heavy-Metal-Platten einzuschmelzen. Eine nicht minder groteske Situation ergibt sich in der zweiten Szene. Micky landet in einer verfallenen Abtei, in deren gruseligem Dämmerlicht betrunkene Hippies randalieren. Micky wehrt sich verbissen seiner Haut und flieht, zerschlagen und zerkratzt — falls er überhaupt mit dem Leben davonkommt. Im dritten Teil des Auftrags gelangt er in ein Aufnahmestudio. Mit Hilfe des vorgegebenen Inventars muß dort der Nummerncode eines Safes ermittelt werden.

Wenn diese drei Nüsse geknackt sind, geht's erst richtig los. Micky wird auf den Wasserplaneten gewarpt (portiert) und in den Körper von Doktor Loach gesteckt. Loach leitet das Projekt, mit dem die Bedrohung durch die Fins abgewendet werden soll. Dazu muß ein Regulator konstruiert werden, dessen fehlende Bauteile Loach finden und korrekt in das halbfertige Gerät einbauen muß. Das Projekt hat allerdings seine Tücken. Denn man muß einen Saboteur enttarnen, ein Museum bestehlen, ein Kernkraftwerk lahmlegen . . . und sich mit typischen Fisch-Problemen herumschlagen. Zum Beispiel in der Bar. Hier wird nicht Schnaps getrunken, sondern Gas geschnüffelt. In bezug auf die Wirkung steht dieses Rauschmittel gewöhnlichem Alkohol in keiner Weise nach.

Der Fall „Fish!“ ist nicht sonderlich schwierig. Allerdings lassen sich die einzelnen Lösungsschritte nicht logisch wie ein Puzzle zusammenfügen. Es bleibt nichts anderes übrig, als Hints aus dem Anleitungsheft einzutippen. Hints bestehen aus aussagelosen Buchstabenkombinationen, die zu jedem beliebigen Zeitpunkt eingegeben werden dürfen. Das Programm entschlüsselt sie und präsentiert unverzichtbare Hinweise oder freche, unnütze Kommentare. Ohne die Hints aus dem Handbuch läßt sich das Spiel nicht lösen, so daß man sie als eine Art Kopierschutz betrachten kann.



Top-Agent Micky ist mit allen Wassern gewaschen. Was liegt näher, als auch Urlaub im nassen Element zu machen.



Die Welt steht kopf — Urlaubsfoto aus dem Goldfischglas.

Grafische Elemente wurden von der Atari-ST-Version übernommen. Die Bilder sind zwar überdurchschnittlich, doch hätte man aus dem Amiga wesentlich mehr herausholen können. Die Amiga-spezifischen Möglichkeiten hat man lediglich für Soundeffekte genutzt. Die Sprachausgabe dagegen funktioniert im guten, alten SAM-Stil (der allerersten Sprachausgabe); mit etwas gutem Willen versteht man die gekrächzten Laute sogar.

In all seinen Stärken und Schwächen betrachtet, gehört „Fish!“ zu den Adventures gehobener Qualität.

(S. Englhart)

Preis: 80 Mark
Hersteller: Magnetic Scrolls
Vertrieb: Ariolasoft, Atlantis

Spaß	8
Grafik	8
Parser	9
Idee	7
Schwierigkeit	6

Emmanuelle

Im Dschungel der Erotik

Rio de Janeiro ist Dreh- und Angelpunkt des Erotikadventures „Emmanuelle" von Tomahawk. In der Stadt am Zukkerhut beginnt das Abenteuer. Hier spielen die meisten Szenen — am Strand, in Bars, am Flughafen, im Hotel. Von Rios Flughafen aus reist der Held in ferne Länder. Akteur ist ein hoffnungslos verliebter Jet-Setter, der seine Traumfrau Emmanuelle in der ganzen Welt sucht, um sie zwecks Heirat nach Europa zu locken. Daß dies kein leichtes Unterfangen wird, stellt sich nach kurzer Spielzeit heraus.

Jung Romeo muß sich nämlich allein auf sein detektivisches Gespür verlassen und versuchen, nützliche Hinweise auf Emmanuelles Aufenthaltsort zu bekommen. Dazu knüpft er Bekanntschaften, stürzt sich in flüchtige Affairen und lebensgefährliche Wagnisse. Aussicht auf Erfolg haben seine Bemühungen jedoch nur, wenn er die Gesetze der Erotik beachtet.

Diese speziellen Regeln des Liebeslebens erfordern außergewöhnliche Einfälle beim Sex, eine ungerade Anzahl von Partnern und wiederholten Verkehr mit den gleichen Partnern. Wann immer eine dieser Regeln artig befolgt wurde, leuchtet eine kleine Figur am unteren Bildrand auf. Doch richteten die tückischen Programmierer es so ein, daß man die entsprechende Figur erst einmal finden muß, bevor man sich an ihr orientieren kann. Figuren und andere Dinge wie der synthetische Liebestrank sind in den entlegensten Winkeln der verschiedenen Schauplätze versteckt. So muß der liebestolle Junggeselle sich beispielsweise die Wasserfälle von Iguacu hinabstürzen, um in den Besitz einer goldenen Statue zu gelangen. Wieviel wertvolle Lebensenergie dabei verbraucht wird, zeigt ein gefülltes Sektglas am linken Bildschirmrand. Aus einer Sektflasche darf nachgeschenkt werden. Wenn die Flasche jedoch leer ist, ist der verliebte Held am Ende seiner Kräfte und auch am Ende des Spiels angelangt. Ebenfalls sparsam sollte er mit seinem Liebespotential umgehen. Denn wer mit jeder verführerischen Dame ins Bett steigt, ist Emmanuelles Liebesbedürfnissen (75 bis 100 Prozent Liebespotential) nicht mehr gewachsen und fliegt allein nach Hause. Lebens- und Liebeskraft sind wichtig — doch dafür kann man sich nichts kaufen. Wichtig ist also, die Finanzen des internationalen Aufreißers im Auge zu behalten, denn Diebe und Gauner bestehlen den Helden, und käufliche Liebesdamen ruinieren das Reisebudget ebenso. Um nicht plötzlich mittellos dazustehen, spielt man ein Spiel im Spiel: Roulette. Im Casino von Rio de Janeiro darf man Jettons setzen und das Glück herausfordern. Pech in der Liebe bringt bekanntlich Glück im Spiel.

Obwohl man das Spiel nur gewinnen kann, wenn man Emmanuelle findet, macht auch das Verlieren Spaß. Denn schließlich lernt man nicht alle Tage so viele Mädchen und so viele Länder kennen wie in diesem Adventure. Für Dialogwitz sorgen flotte Sprüche. „Hast du die Erlaubnis, alleine auszugehen?" oder „Baby, wollen wir eine Nummer zusammen abziehen?" „Emmanuelle" ist ein sehr gelungenes Adventure, das auf erwachsene Computerfreaks zugeschnitten ist. Wer allerdings pornografisch angehauchte Szenen erwartet, ist im falschen Spiel. Dieses Erotik-Adventure ist weder pornografisch noch vulgär. Vielmehr soll es mit bunten Bildern, Erotik, witzigen Wortwechseln und komplexer Handlung Verstand und Phantasie anregen.

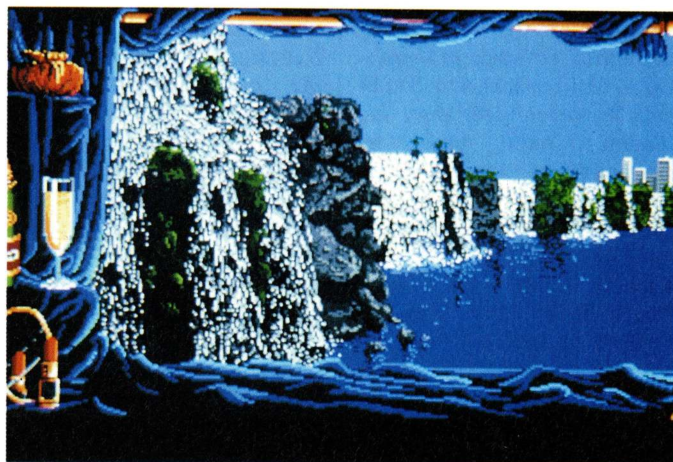
(mms)

Preis: 65 Mark
Hersteller: Tomahawk
Vertrieb: Bomico

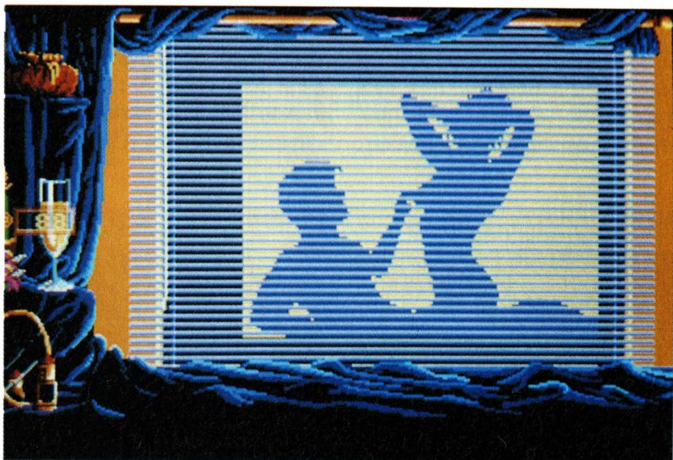
Spaß	10
Grafik	8
Sound	5
Idee	10
Schwierigkeit	9



Weiß die Dame etwas über Emanuelle? Beim zwanglosen Gespräch versucht man das herauszufinden.



Im Namen der Liebe stürzt sich der Held 72 Meter von den Wasserfällen des Iguacu hinunter.



Manche Informationen erkauft der Held mit Liebeskraft. Das kann ihn teuer zu stehen kommen, denn Emanuelle ist anspruchsvoll beim Geschlechtsverkehr. Sie will alles.

Elite

Donauwalzer

Vier Jahre lang warten Amiga-Besitzer nun darauf, Weltraumhandel treiben zu dürfen. Denn um diese Zeitspanne sind ihnen die C64-Spieler voraus. Endlich ist „Elite", das Vorbild vieler Handelssimulationen, für den Amiga erhältlich. Jetzt werden strategisches Geschick, schnelle Reaktion und ausdauernde Daumenmuskeln gebraucht.

Kurz nach dem Start stellt sich heraus, daß fast keine Ausrüstung vorhanden ist. Die wird für eine steile Karriere als Weltraumhändler jedoch unbedingt benötigt. Zu Spielbeginn besteht die dürftige Ausstattung aus einem schwachen Impuls-Laser, drei Suchraketen. Treibstoff für sieben Lichtjahre, und hundert Credits und dem Raumschiff — Modell Cobra MK III.

Ziel aller Aktivitäten ist der Titel „Elite". Um diese Auszeichnung zu erhalten, gibt es zwei Möglichkeiten. Den Weg der Legalität und den der Piraterie. Wer sich für Gesetzestreue entscheidet, verdient sich seinen Lorbeer durch Handel, Erzabbau und Kampf gegen Piraten. Es steht dem Spieler aber auch offen, selber Pirat zu werden. Das bedeutet, mit verbotenen Waren zu handeln und andere Raumschiffe zu plündern. Anfänger sollten sich nicht der Piraterie verschreiben, da dieses Handwerk einige Erfahrung und Ausrüstung erfordert. Doch auch der legale Weg birgt ungeahnte Schwierigkeiten. Denn wer gewinnträchtige Handelsgeschäfte abschließen will, muß sich nicht nur mit der wirtschaftlichen, sondern auch mit der politischen Struktur eines Planeten auskennen. Politische Wirren machen eine Annäherung an den Planeten zur unberechenbaren Gefahr. Zum anderen sollte man sich über den technologischen Level eines Planeten informieren, da Preis, Angebot und Nachfrage in hohem Maße davon abhängen. Dann ist sorgfältig abzuwägen, ob bei dem aktuellen Entwicklungsstand überhaupt Nachfrage nach den mitgeführten Waren besteht. Doch muß ein Weltraumhändler sich nicht nur mit ökonomischen Belangen auskennen, sondern auch für den Kampf gegen Piraten gewappnet sein.

Im Kampf um den Rang „Elite" kann man, sofern die richtige Ausrüstung vorhanden ist, auch andere Möglichkeiten wahrnehmen. Wer beispielsweise einen Bergbau-Laser — Typ Felsknacker 15 — und einen Brennstofflader besitzt, kann sich mit dem Abbau von Asteroiden zusätzliche Credits verdienen.

Ebenfalls vorteilhaft für Kontostand und Status ist die Zerstörung von Piratenschiffen. Geld, das man auf diese anstrengende Art und Weise verdient hat, wird sinnvollerweise in Angriffs- und Verteidigungssysteme investiert. Alle notwendigen Waffen sind auf den unterschiedlichen Planeten erhältlich. Je nach technischem Entwicklungsstand gibt es dort verschiedene Laser-Modelle, EDM-Systeme, Raketen, Bomben, Andock-Computer und anderes Kriegsgerät.



Wie wär's mit einem Abstecher nach Lave? Es soll dort nette Casinos geben.



Raumschiff Cobra zu Spielbeginn: Die Tankfüllung reicht für sieben Lichtjahre; nur die Bewaffnung ist dürftig.

Komplexität ist Trumpf bei „Elite". Die Vielfalt der Waren und Waffen, der Feinde und Freunde, der politischen Systeme und der Sternensysteme scheint schier unerschöpflich. In acht Galaxien mit je 250 bekannten Planeten werden unablässig Absatzmärkte erforscht und verteidigt. Selbst wenn es dem Spieler gelungen sein sollte, den begehrten Titel „Elite" zu erlangen, bietet das Spiel noch Spezialmissionen, die auch einen elitären Ökonomen herausfordern.

Die ausgefüllte Vektorgrafik ist zwar durchaus von gehobener Qualität, doch wurde das grafische Potential des 16-Bit-Rechners nicht ausgeschöpft. Verwirrende Farbenvielfalt und phantasievoll dargestellte Planetenbewohner zusammen mit dem strategischen Anspruch machen „Elite" zu einem aufwendigen Spiel. Der Sound steht hinter konzeptionellen und grafischen Qualitäten zurück. Lediglich beim Andocken ist der legendäre Donauwalzer zu hören. Abgesehen davon gibt es keine nennenswerten Soundeffekte. Das Spiele-Handbuch ist tadellos, es erklärt alles, was zum Verständnis des Spiels notwendig ist.

„Elite" ist eine gelungene Mischung aus Aktion und Simulation. Ausnahmsweise ist dies ein Spiel, das über Monate hinweg nicht an Reiz verliert.

(U. Loos/A. Peters)

Preis: 70 Mark
Hersteller: Firebird
Vertrieb: Ariolasoft

Wertung	Punkte
Spaß	10
Grafik	8
Sound	4
Idee	9
Schwierigkeit	9

Editor hinter der Elite-Kulisse

Wer hätte das gedacht: Auf der Elite-Diskette befindet sich schon ein Scenery-Generator. Nach mühevoller Arbeit ist es gelungen, das System des im Spiel verborgenen Character-Generators zu entschlüsseln. Um den Editor zu starten, muß bei der Codeabfrage statt des ersten geforderten Paßwortes „sara" eingetippt werden, anschließend, wie gewohnt, Paßwort eingeben.

Wenn sich ein Originalprogramm der Firma Acornsoft in Ihrem Laufwerk befindet, läßt sich mit der Taste „*" auf dem rechten Zahlenblock der Editor

„Hacker V1.0" aufrufen. Hierbei handelt es sich um einen bedienerfreundlichen Maschinensprachmonitor. Dieser arbeitet mit dem Hexadezimalsystem. Jetzt hat man die Möglichkeit, durch Ändern einzelner Zahlenwerte verschiedene Spielzustände zu erzeugen. Zum Beispiel ist es möglich, die Einstufung zu erhöhen oder die Finanzen zu manipulieren. Nachfolgend eine Zusammenstellung der zu ändernden Bytes. Zur Ware: Die Menge der Ware ist in ein System aufgegliedert, das heißt, das erste Byte sind die Tonnen, das zweite die Kilogramm, das dritte Byte, Gramm. Beispiel: Wird das Byte 5b mit FF belegt, so erhalten sie 256 Gramm Drogen.

Vorsicht ist bei dem Byte des Sprits (1e, 1f) angebracht. Bei einem Wert über 46 hängt sich das System auf.

Das gleiche Problem ergibt sich bei der Manipulation der Zeit. Deshalb sollten Sie niemals die Maximalwerte überschreiten. Um nicht den Spaß an diesem wirklich exzellent programmierten Spiel zu verlieren, sollten nur geringfügige Veränderungen vorgenommen werden.

(Guido Bär/Ralf Böhm)

Byte-Änderungen machen stark. Wer es aber übertreibt, erreicht das Gegenteil. ▶

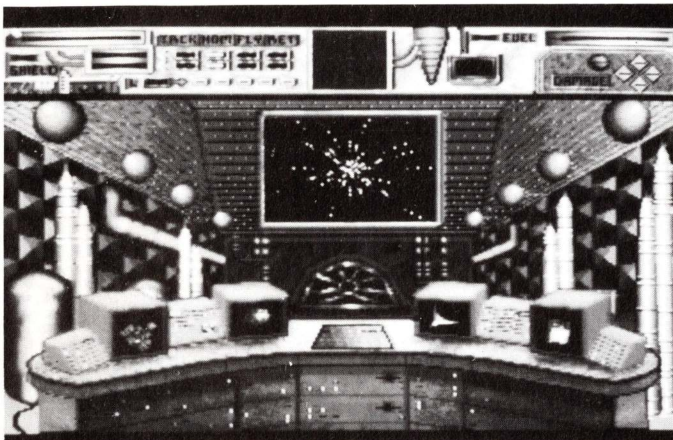
```
Byte 00,01,02,03,04,05,06,07        =  Name des Commanders
Byte 0c,0d                          =  Wahl der Galaxis
Byte 0e,0f,10,11,12,13,14,15        =  Wahl des Planeten
Byte 16                             =  Options
Byte 18,19,1a,1b                    =  Credits (Geld)
Byte 1e,1f                          =  Sprit
Byte 30,31,32,33,34,35,36,37
     38,39,3a,3b,3c,3d,3e,3f        =  Ausrüstung
Byte 40,41,42,43                    =  Nahrung
Byte 44,45,46,47                    =  Textilien
Byte 48,49,4a,4b                    =  Radioaktive Stoffe
Byte 4c,4d,4e,4f                    =  Sklaven
Byte 50,51,52,53                    =  Spirituosen
Byte 54,55,56,57                    =  Luxusgüter
Byte 58,59,5a,5b                    =  Drogen
Byte 5c,5d,5e,5f                    =  Computer
Byte 60,61,62,63                    =  Maschinen
Byte 64,65,66,67                    =  Metalle
Byte 68,69,6a,6b                    =  Waffen
Byte 6c,6d,6e,6f                    =  Felle
Byte 70,71,72,73                    =  Mineralien
Byte 74,75,76,77                    =  Gold
Byte 78,79,7a,7b                    =  Platin
Byte 7c,7d,7e,7f                    =  Edelsteine
Byte 80,81,82,83                    =  Exotica
Byte 84,85,86,87                    =  Medikamente
Byte 88,89,8a,8b                    =  Pelztiere
Byte 8c,8d,8e,8f                    =  Thargoidische Dokumente
Byte 90,91                          =  Führungszeugnis
Byte 92,93,94,95                    =  Punkte
Byte 97                             =  Einstufung
Byte a2,a3,a4                       =  Zeit
```

Pioneer Plague

Sonden drehen durch

Mit „Pioneer Plague" wurde das erste Strategiespiel mit Hold-and-Modify-Grafiken realisiert. Endlich mal was Neues! Nicht so neu dagegen ist die Problematik, die hier in eine spielerische Form umgesetzt wurde – nämlich sich verselbständigende Computerintelligenz. Im Spiel vollzieht sich dieser beängstigende Vorgang bei Pioniersonden, deren ursprünglicher Zweck es war, Planeten für die Kolonisierung vorzubereiten. Jahrelang arbeitete das Steuerprogramm der Sonden einwandfrei. Die elektronischen Pioniere flogen zu fernen Wüstenplaneten und verwandelten deren Oberflächen in Parklandschaften, Städte und Fabrikanlagen. Damit aber nicht genug! Gemäß ihrer Mission,, errichteten die Pioniersonden Vervielfältigungsbasen. Dort produzierten sie Kopien ihrer selbst und übertrugen ihr Steuerungsprogramm identisch auf die Clones. Jetzt hat sich beim Kopieren ein fataler Fehler ins Programm geschlichen. Die Sonden tun plötzlich genau das Gegenteil von dem, was sie eigentlich tun sollten. Sie nehmen wieder Kurs auf die Erde und wandeln alles um, was ihnen in die Quere kommt. Wenn sie die Erde erreicht haben, werden sie nicht nur die Welt zur Wüste machen, sondern auch Mensch, Tier und Pflanze zerlegen und zu leblosen Strukturen synthetisieren. Noch haben die zerstörerischen Sonden die Erde nicht erreicht. Sollte aber niemand die Gefahr aufhalten, wird der blaue Planet zur unbelebten Geröllkugel. Hier muß der Spieler eingreifen. Mit seinem Raumschiff fliegt er in die Umlaufbahnen der von Sonden besetzten Planeten. Dann beginnt der Kampf gegen die einstigen Diener des Menschen. Mit gezieltem Bombardement versucht der Spieler die unheilvollen Brutstätten der Sonden zu zerstören. Das wird dem Spieler jedoch schwer gemacht. Die falsch programmierten Pioniere haben zahlreiche Abwehreinrichtungen gebaut, um ihr zerstörerisches Werk zu schützen. Der irdische Retter muß nun wahre Multitasking-Fähigkeiten beweisen, denn er muß sich gegen alle möglichen Gefahren gleichzeitig wehren. Artilleriegeschosse drohen das kleine Raumschiff zu zerschmettern. Sondengelenkte Jagdflugzeuge verfolgen es in Scharen. Raketen rasen durch die dünne Luft. Und schwebende Minen zwingen zu riskanten Ausweichmanövern. All dies ereignet sich in einem Ambiente berauschender Farbenvielfalt. Dazu ertönt Stereosound. „Pioneer Plague" ist rundum gut — wenn nur das Scrolling nicht so zähflüssig wäre! Das Flugzeug ist schneller als das Scrolling, so daß man am Bildschirmrand unfreiwillige Pausen einlegen muß. Unerfreulich für alle Nichtanglisten ist das englische Handbuch. Umständlich wird es, wenn der Bordcomputer ausfällt, weil dann mit Hilfe des Emergency Manual Navigation Kit (EMNK) manuell gesteuert werden muß. Solche Macken an der Dokumentation werden im nächsten HAM-Spiel hoffentlich ausgemerzt sein. Jedes HAM-Programm wird jedoch Schwächen haben, da die Grafik wertvollen Speicherplatz verbraucht, der für andere Funktionen fehlt.

(R. de Flers)



Im HAM-Modus angefertigte Grafiken zeichnen sich durch schillernde Buntheit aus.

Preis: 80 Mark
Hersteller: Mandarin/Terrific Software
Bezugsquelle: Müller

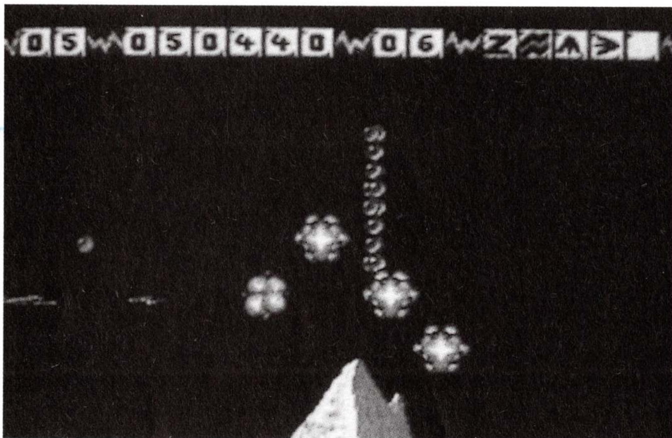
Spaß	7
Grafik	10
Sound	9
Idee	6
Schwierigkeit	10

Excalibur

Ohne Schwert ins All

Es geht die Sage, daß im 6. Jahrhundert nach Christus der Einfluß Avalons auf den britannischen Thron mit dem magischen Schwert Excalibur verteidigt wurde. Diese Geschichte mag man glauben oder nicht. Man liegt auf jeden Fall dann falsch, wenn man annimmt, im neuen Kingsoft-Spiel „Excalibur" auf den Spuren von König Artus zu wandeln. „Excalibur" folgt der inflationären Tendenz zu Weltraum-Ballerspielen. Ein kleines Raumschiff fliegt über die Oberfläche eines bergigen Planeten und schießt auf alles, was sich regt — so das Spielprinzip von „Excalibur". Im Schutz von USOs (undefinierbaren schwebenden Objekten) nähern sich unablässig feindliche Formationen. Da gibt es die verrücktesten Fluggeräte, vom vertrauten Raumschiff bis zum merkwürdig hüpfenden Edelstein. Der Abschuß einer Formation wird mit Bonuswaffen belohnt. Auf diese Weise lassen sich Feuerkraft, Beweglichkeit und Schutzschild des Raumgleiters verstärken. Nach jedem Absturz beginnt die mühselige Bonusjagd aufs neue. Je Spiel darf man drei Schiffe zu Schrott fliegen, und wer genügend Punkte sammelt, bekommt zusätzliche Bonusschiffe (ab 10 000 Punkte). „Excalibur" unterscheidet sich nicht von anderen Spielen dieser Kategorie und ist trotz seines magischen Namens auch nicht fesselnder als diese. Leicht ist das Prinzip durchschaut. In puncto Grafik hat man sich viel Mühe gegeben, die Illusion von Raumtiefe heraufzubeschwören. Abwechslung wird jedoch wenig geboten; der Hintergrund verändert sich lediglich in der Farbe. Mehr Phantasie beweist die Titelmelodie. Alle anderen Spielgeräusche klingen dürftig; Bonussignale tönen wie die Etagenklingel eines Fahrstuhls. Interessant ist die Implementierung der Schwerkraft. Das Raumschiff bewegt sich gemäß der Anziehungskraft gen Boden, wenn nicht per Joystick eine Aufwärtsbewegung gesteuert wird. Nicht Schnelligkeit am Feuerknopf entscheidet dieses Spiel, sondern das Augenmaß eines trefflichen Schützen.

(mms)



Blasenförmige Aliens wirbeln in Angriffsposition.

Preis: 40 Mark
Hersteller/
Vertrieb: Kingsoft

Spaß	5
Grafik	5
Sound	3
Idee	5
Schwierigkeit	5

Tanglewood

Endlose Probleme

In der Septemberausgabe wurde es angekündigt. Nach einigen Schwierigkeiten ist es jetzt da: „Tanglewood" von Microdeal.

Das Abenteuer geht schon beim Lesen der Gebrauchsanleitung los. Man gewinnt den Eindruck, daß sie von einem Australopithecus (Primat des Tertiärs) vom deutschen über das lateinische in das babylonische Sprachgewirr und zurück in das Deutsche übersetzt wurde. „Tanglewood" ist weder Reflex- noch Ballerspiel. Es fordert zur Lösung von Problemen heraus. Am Ausgangspunkt des Spieles ist der Spieler Besitzer der Schürfrechte auf dem Planeten Tanglewood. Wegen der kostbaren Mineralvorkommen erschleicht sich aber eine große Minengesellschaft die Schürfrechte. Der Spieler muß nun die Urkunden innerhalb von zehn Tagen finden, um vor Gericht seinen rechtmäßigen Anspruch durchsetzen zu können. Einerseits muß man sich mit den Ureinwohnern von Tanglewood gut stellen und andererseits mit den fünf zur Verfügung stehenden Oberflächenfahrzeugen herumärgern. Die Fahrzeuge sind entweder mehr oder minder Wracks oder sie können nur unter bestimmten klimatischen Bedingungen und an definierten Tageszeiten eingesetzt werden. Da kann es schon mal passieren, daß ein Fahrzeug die gesammelten Daten und Objekte verliert oder außer Gefecht gesetzt wird. Dann heißt es mit einem anderen Fahrzeug zu dem Wrack zu gelangen, um Pannenhilfe zu leisten. Dies wird sehr erschwert, da man keinen globalen Überblick über den Planeten hat.

Man sucht also kreuz und quer, auf und unterhalb der Planetenoberfläche nach den Urkunden. Bei diesem Unterfangen beginnt das Spiel langatmig zu werden. Hin und wieder kommt man in bestimmte Gegenden, die durch schöne Grafiken versöhnen, bevor es mit der Suche weitergeht. Ist man endlich in die unterirdische Zentrale der Minengesellschaft vorgestoßen und damit den gesuchten Urkunden ein Stück näher, ist entweder die Zeit abgelaufen oder man wird vom Sicherheitssystem außer Gefecht gesetzt und muß von vorne beginnen. „Tanglewood" ist ein Adventurespiel, das einen vor ein großes Problem stellt und dessen Lösung viel Zeit und Durchhaltevermögen erfordert. Es wird am besten mit der Joystick/Mouse-Kombination gespielt.

(R. de Flers)



Landschaftsschutz auf Tanglewood: naturnahe Bauweise im Fliegenpilzstil.

Preis: 60 DM
Hersteller: Microdeal
Vertrieb: Rushware

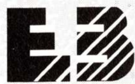
Spaß	4
Grafik	6
Sound	2
Idee	6
Schwierigkeit	9

Amiga Loads Faster

Das komplette Antriebssystem für die Harddisk:
Sicherer durch CheckDrive - Schneller mit FastFileSystem
Ergiebiger mit RLL-Controller - Billiger, weil auch
defekte Platten komplex verwaltet werden können.

A.L.F.

macht der Harddisk *Beine*



OLIVER KASTL

ELABORATE BYTES

Schleißheimer Str. 205 a D-8000 München 40
TEL 089 - 307 21 71 oder 308 41 52 FAX 089 - 307 17 14

SOFTWARE AUS DEUTSCHLAND



Lübecker Straße 10
2320 Plön/Holstein
Telefon 0 45 22/13 79

Distributoren:
BRD: Casablanca GmbH,
Mehringskamp 9, 4630 Bochum 5

Österreich: Intercomp, Gschwend 163,
A-6932 Langen b. Bregenz

Weitere AMIGA-SOFTWARE in Vorbereitung! Alle Programme laufen auf allen AMIGA-Modellen! Bestellungen schriftlich oder telefonisch unter **04522/1379.** Gegen 1,90 DM in Briefmarken erhalten Sie ausführliche Produkt-Infos. Versand gegen Vorkasse oder per Nachnahme zuzüglich 5,— DM für Porto und Verpackung.



KREUZ AS
- Sehr guter Poker-Spielautomat
- Nach dem Original-Spielhallen-
- Hit „Herz As"
- Sehr gute Grafik und
- dig. Sound

Best.-Nr. S 02 002 88
Preis: 79. — DM

AMIGA ROULETTE
- Tolle Grafik
- bis zu 4 Spieler
- hoher Spielspaß
- alle Roulette-Regeln werden
- berucksichtigt
- dt. Anleitung

Best.-Nr.: S 01 001 88
Preis: 69. — DM

BUNDESLIGA - MANAGER
- Werden Sie zum Manager
- Ihres Fußballvereins
- Verhelfen Sie Ihrem Verein zu
- Siegen und Geld
- Sehr gute Wirtschafts-
- Simulation
- hoher Spielspaß
- bis zu 4 Spieler

Best.-Nr. S 02 001 88
Preis: 69. — DM

NEUE AMIGA - SOFTWARE

Tausende von Kambodschanern wurden im Indochinakrieg zu Krüppeln, als die USA ihre Bombenangriffe auf ihr Land ausweiteten. Bis heute, Jahrzehnte später, erhielten sie weder nationale noch internationale Hilfe. Jetzt will sich eine spezielle Organisation für Kriegsbehinderte (OHI) zusammen mit amerikanischen Quäkern um diese armen Menschen kümmern: Sie bilden Einheimische in der Anfertigung von einfachen Prothesen und in der Durchführung von Rehabilitationsmaßnahmen aus.

BROT FÜR DIE WELT - Spenden helfen bei der Finanzierung.
Postfach 10 11 42 · 7000 Stuttgart 10

Brot für die Welt
Postgiro Köln 500500-500



Peter Biet Computerdesign
Georg-Fischer-Straße 5, 6415 Petersberg 2
Telefon 06 61/60 12 63





PAL-RGB-MULTIPROZESSOR 2000 (Version I, II und III)

Die neue Gerätegeneration, mit noch mehr Einsatzmöglichkeiten. Durch die enorme Vielseitigkeit verknüpft mit guten Leistungsmerkmalen und einem bisher unschlagbaren Preis-/Leistungs-Verhältnis stellt dieses neukonzipierte Gerät einen Meilenstein im Bereich der Desktop-Videobearbeitung mit dem Amiga dar.

Einsatzbereiche:

1.) Elektronischer **RGB-Farbsplitter** (automatisch und manuell!). Separate Regelmöglichkeiten in den Bereichen **Helligkeit, Kontrast, Farbsättigung** und die einzelnen **RGB-Intensitätswerte.**
Damit ist es auch mit **DIGI-VIEW** und einer Farbcamera oder einem Videorecorder möglich, absolut professionelle Ergebnisse zu erzielen!
2.) **Video-Colorprozessor:** dient zur Veränderung bzw. Verfremdung des anliegenden Farbvideosignals in o. g. Bereichen zwecks Überwachung, Kontrolle und Überspielung.
3.) **Videoüberspielverstärker:** mit Level und Konturregelung.
4.) **RGB-PALUMWANDLER:** erzeugt aus dem Computer-RGB-Signal ein PAL-Videosignal. Ebenfalls in o. g. Bereichen regelbar! (Ersetzt damit alle herkömmlichen PAL-Videokarten.)

neu 19-Zollgehäuse mit neuem Design, Digitastern, Anzeigendisplay und Chinchbuchsen.
neu Mehrfachüberspielmöglichkeiten auf bis zu drei Videorecorder gleichzeitig.
neu Anschlußmöglichkeiten eines Multisynchronmonitors als Videomonitor über RGB-Ausgang.
neu Einspeisung des neuen **Super VHS** Signals und Umwandlung in ein normales FBAS-Signal. (Vollständig regelbar ohne Qualitätsverlust!)
neu DIGI-VIEW-Digitizer kann direkt intern eingesteckt werden. (Stromversorgung durch eigenes Netzteil, keine externe Kabelverbindung mehr nötig und Druckerumschaltmöglichkeiten!)
Unterschiede der Versionen und Preise auf Anfrage

DVS-2000 System

Das bewährte Realtimefixing Digitizersystem in voller PAL- und Overscanauflösung.

Lieferumfang:
— **Digitizer** (alle Auflösungen incl. HAM 2-4096 Farben in PAL!)
— **Digitaler VHS-Videorecorder** (eigenständige Nutzung möglich!) (kann auch durch einen Super-VHS-Recorder ersetzt werden!)
— **PAL-RGB-MULTIPROZESSOR 2000** Version III **zusammen nur 2598.— DM** (auch einzeln!)
Demo-Dia-Show (DVS-2000 und PAL-RGB-MULTIPROZESSOR 2000) **10.— DM**

PAL-RGB-GENLOCK

Das neue Multifunktionstalent mit eingebautem Audiomischer und **Fernbedienungseinheit.** Separate RGB- und FBAS-Ausgänge. Stufenloses Ein- bzw. Ausblenden von Computer- und Videosignalen über separate Regler möglich. **1098.— DM**
NEU: Panasonic Super-VHS-Videorecorder (Hifi + Nachvertonung!) **2998.— DM**
NEU: Panasonic Camcorder MC-10 **3198.— DM**

Computer

Wir installieren, warten, reparieren – schnell, preiswert und gut!

Schneider
ATARI
commodore

RAT UND TAT finden Sie vom Norden bis zum Süden – an 70 Standorten:

Hotline Bereich Nord (040) 2201913
Hotline Bereich Mitte (0201) 35923
Hotline Bereich Süd (08165) 74220
Hotline Berlin (030) 6846057-9

Unser Service





Technischer
Kundendienst

Wir reparieren für Sie

Fortsetzung von Seite 72

Disks, so daß auch Sie, sollte der Wunsch vorhanden sein, genauso eine Startup-Sequence aufbauen können. Dabei ist darauf zu achten, daß alle Programme nicht mittels des normalen „run"-Kommandos, sondern mittels „runback" ausgeführt werden, da sich sonst das große Fenster am Ende nicht wieder schließt. Am Ende der Startup-Sequence werden dann noch residente Kommandos der 1.3 eingebunden sowie der neue Tastatur-Handler ge„mountet" und ein Shell geöffnet. Nun kehrt die Ausführung wieder in die letzte Zeile der echten Startup-Sequence, der auf der Boot- Diskette, zurück, wo das Fenster, wie oben erwähnt, sofort geschlossen wird. Das System ist „hochgefahren", und alle benötigten Utilities sind von der Festplatte aus gestartet worden. Messen Sie einmal den Zeitgewinn gegenüber einer Version, in der alle diese Sachen noch auf Diskette „zusammengekratzt" werden müßen.

Ein Manko hat dieses „Mammutwerk" jedoch noch. Tatsächlich gebootet wird immer noch von Diskette, und da dieser Umstand unter Verwendung der Kickstart 1.3 zwar nicht umgangen, jedoch gelindert werden kann, soll auch diese hier gezeigt werden. Dazu wird das Verhalten der RAD-Disk benutzt, die automatisch bootet, wenn keine Diskette eingelegt ist. Dazu muß natürlich beim ersten Mal die Boot-Diskette einmal ins RAM kopiert werden. Speichern Sie auf die tatsächliche Boot-Diskette wirklich nur die Dateien, die für den Anfang benötigt werden, paßt so die gesamte Diskette in ein paar Spuren der RAD, was Sie wiederum in der Mountlist feststellen können (Highcyl). Das spart viel Speicher!

Nun kann an den Anfang der Startup-Sequence auf der Boot-Disk noch der in Tafel 5 dargestellte „Zusatz" geschrieben werden. Hier wird in einem einfachen, aber effektiven Verfahren festgestellt, ob eine RAD-Disk vorhanden ist, von der ja dann wahrscheinlich auch gebootet wurde.

Dazu wird zuerst die Fehlerabbruchstufe hoch gesetzt, damit die Abarbeitung auch bei einem auftretendem Fehler fortschreitet. Nun wird versucht das von mir umbenannte RAD-Gerät mittels „assign" zu löschen, was natürlich nur funktioniert, falls das RAD-Gerät noch nicht existiert. Ist das der Fall, wird gefragt, ob die RAM-Disk eingerichtet werden soll und wenn ja, wird mit dem Kopiervorgang begonnen. All das geschieht nur, wenn KEINE RAD-Disk vorhanden war, also von ihr auch nicht gebootet werden konnte. Ist der Kopiervorgang beendet, kann die Diskette aus dem Laufwerk genommen und ein Reset erzeugt werden. Nun müßte die Festplatte noch schneller und vollkommen ohne Diskettenzugriff „hochfahren".

(Ottmar Röhrig)

```
echo "Assigning to dh0:..."
assign SYS: dh0:
assign L: SYS:l
assign LIBS: SYS:libs
assign FONTS: SYS:fonts
assign S: SYS:s
assign DEVS: SYS:devs


echo "Mounting partitions..."
mount dh1:
mount dh2:
mount dh3:
mount dh4:


;run >NIL: SetAlert                              ;patch system

dh0:tools/setfont clean 8                                  ;font verändert
echo "Changed Font..."


echo " "
echo "Systemtime & -date set by internal clock: " noline
setclock opt load
            ;settime


date
echo " "
dh0:tools/hello "!  Enjoy and have fun..." ;begrüßen
echo ""
echo " "


echo "Installing ASDG-Recoverable-RAM-Disk..."
mount vd0:
copy sys:ASDG-RAM.info vd0:Disk.info

loadwb -debug
wait 5


echo "Installing GOMF2.1..."
sys:system/gomf2.1
echo "Installing FastFonts..."
FF >NIL: -0                                     ; start FastFonts

dh0:System/SetMap d

;Sys:System/FastMemFirst                        ; move C00000 memory to last in

BindDrivers


echo "Setting Environment, Assigns and Path..."
MakeDir Ram:T                                   ; set up logical T: directory
Assign T: ram:T
MakeDir Ram:Env                                 ; set up ENV: directory
Assign ENV: ram:Env
;path sys:utilities sys:system sys:tools ram: sys: s: dh2: vd0: m2:CompilerSt


Assign system: sys:system
Assign util: sys:utilities
Assign to: sys:tools


;----------------------------------------------------------------


Assign extras: dh2:basic       ;Program Assigns!!!
Assign m2: dh2:m2
path m2:CompilerStuff add

assign LOVLY:  dh2:XCAD/lovly
assign POVLY:  dh2:XCAD/povly
assign MODS:   dh2:XCAD/mods
assign XFONTS: dh2:XCAD/xfonts
assign XCDATA: dh2:XCAD/xcdata

assign PPage:         dh2:PPage
assign PPageUtil:     dh2:PPage


;----------------------------------------------------------------


;echo "New pointer..."
;dh0:system/pointer dh0:system/checkmark -16 -0
echo "Installing ViCheck..."
runback >nil: system:ViCheck
echo "Installing MClk..."
runback >nil: system:mclk
echo "Installing QuickMouse..."
runback >nil: system:quickmouse
echo "Loading FunctionKeys...(activate with right-alt-HELP)"
runback >nil: util:FuncKey util:func.key
echo "Installing PopCLI...(activate with left-amiga-ESC)"
run >nil: system:popcli 300 newshell


resident CLI L:Shell-Seg SYSTEM pure           ; activate Shell
resident c:Execute pure                        ; pre-load the Execute command

mount newcon:                                  ; mount the improved console handle

newshell
```

Tafel 4: Die Disk-Startup-Sequence ruft die eigentliche von der Festplatte auf.

Dragon's Lair

Im Nest des Drachen

Fabelwesen wie Drachen und Seeungeheuer faszinieren Fantasy-Freunde. All jenen, die in einer Fantasy-Story mitwirken wollen, bietet Readysoft das komplexe Bildschirmabenteuer „Dragon's Lair". Schöpfer dieser Mischung aus Trickfilm und Spiel ist Don Bluth, der ehemalige Chef der Disney-Studios. Daß hier ein Profi am Werke war, zeigen die aufwendigen und perfekten Animationen. Die erforderlichen 130 MB Grafik- und Animationsdaten füllen sechs Disketten.

Der Spieler dieses Mammutwerkes bekommt die Rolle Dirks des Wagemutigen. Dirk will die holde Prinzessin Daphne aus den Klauen eines grausamen Drachen befreien. Doch dazu muß er das Versteck des Drachen erst einmal finden. Bei seiner Suche im Drachenschloß gerät er in ein Abenteuer nach dem anderen.

Kaum hat Dirk die Zugbrücke des Schlosses betreten, fällt er in ein Loch und kann sich gerade noch an dessen Rand festhalten. Für einen kräftigen Mann wie Dirk wäre es kein Problem, sich mit einem Klimmzug zu retten – würden da nicht kräftige Fangarme aus der Tiefe des Burggrabens nach ihm greifen und versuchen ihn ins Wasser zu ziehen. Es gibt nur eine Lösung. Dirk wartet, bis die schleimigen Fangarme in Reichweite kommen und zückt das Schwert. Ein Druck auf den Feuerknopf oder auf die Taste 0 löst den Schwerthieb aus. Das Ungeheuer weicht zurück. Blitzschnell muß der Spieler reagieren und den Joystick nach vorn oder die Taste 8 drücken. Dirk kriecht auf die Brücke, steht auf und rennt ins Burginnere – das vielarmige Ungeheuer greift ins Leere.

Die nächste Szene ereignet sich im Vorraum der Burg. Dirk entdeckt eine Flasche mit der Aufschrift „drink me". Das klingt verdächtig! Falls der Ritter die Einladung annimmt, wird er einen qualvollen Gifttod sterben. Um das zu verhindern, muß Dirk sich an einer bestimmten Stelle nach rechts wenden und durch eine Holztür ins Freie hasten. Er steht wieder auf der Zugbrücke. Mit den lästigen Fangarmen wird er diesmal spielend fertig. Dann geht es eine Treppe hinunter, die aufgrund ihres Alters oder durch Zauberkraft zerbröckelt. Vor und unter Dirks Füßen brechen große Teile aus der Treppe. Mit weiten Sprüngen im rechten Moment gelingt es ihm, den Keller lebend zu erreichen.

Von Verschnaufen kann aber auch hier keine Rede sein. Unter der Decke des Gewölbes ringelt sich eine grüne Riesenschlange. Sobald sie angreift, muß Dirk ihren Schwanz abschlagen und mit einem mächtigen Satz zum Waffenschrank springen. Da ist eine Tür. Aber die Schlange blockiert den Fluchtweg, und es kommt zum ungleichen Zweikampf. Dies ist die schwierigste Sequenz der zweiten Diskette. Immer wieder wird der Held zu Brei gequetscht. Es kostet viel Zeit und viele Heldenleben, herauszufinden, was wann zu tun ist. Wer es schafft, bekommt endlich den ersten Drachen zu Gesicht. Ahnungslos nähert sich Dirk einem großen Eisenkessel mit schleimiger grüner Brühe darin. Plötzlich fließt der ekelhafte Schleim aus dem Kessel und wird zum Drachen. Dirk benutzt fleißig das Schwert.

Dann geht Dirk durch einen Hohlweg mit glatten Wänden. Bunte Kugeln rollen hindurch. Schließlich wälzt sich eine große schwarze Kugel auf den Helden zu. Davor rettet ihn kein Schwert, nur Ausweichen. Bei der rasanten Bootsfahrt über vier Strudel und vier Wasserfälle ist mit Waffen auch nichts auszurichten.

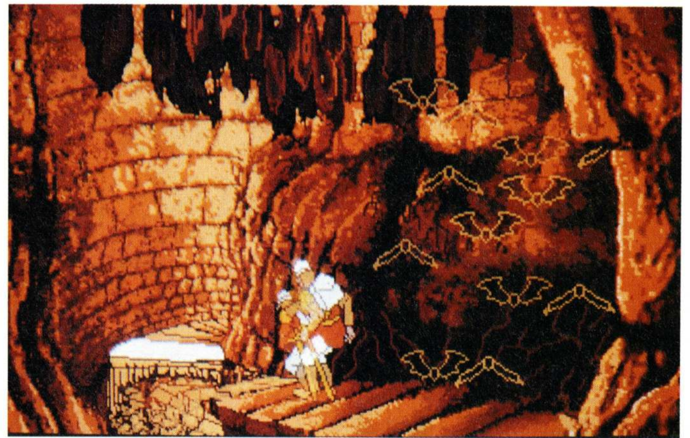
Im schachbrettartig gemusterten Saal muß Dirk Stromschlägen ausweichen, die ein Riese auf ihn schleudert. Gleichzeitig muß er darauf achten, nicht auf leuchtende Schachfelder zu treten. Den paradoxesten Kampf führt Dirk jedoch gegen einen flinken Schattenkrieger.

„Dragon's Lair" ist kein Belastungstest für Joysticks. Der Spieler hat wenig Möglichkeit, das Abenteuer zu beeinflussen. Es kommt in diesem Spiel nur auf das Timing an. Wer nicht im rechten Moment und an der richtigen Stelle auf den Feuerknopf drückt, hat das Leben des Trickfilmhelden verwirkt. Nach drei Fehlreaktionen ist das Spiel zu Ende. Wenn Dirk gegen alle Bösewichter der sechs Disketten gekämpft und den Drachen „Singe" besiegt hat, wird seine Kühnheit mit der Liebe der Prinzessin Daphne belohnt.

(mms)



Schleimige Brühe formt sich zum Drachen. Ein Schwerthieb beendet die Metamorphose.



Kaum hat Dirk eine Stufe berührt, poltert die halbe Treppe in den Abgrund.

Preis: 120 Mark
Hersteller: Readysoft
Vertrieb: Casablanca, Leisuresoft

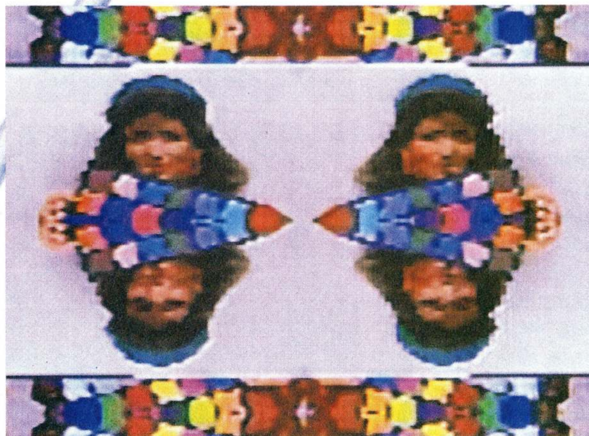
Spaß	7
Grafik	10
Sound	9
Idee	9
Schwierigkeit	10

VORSCHAU

3/89 Mai/Juni

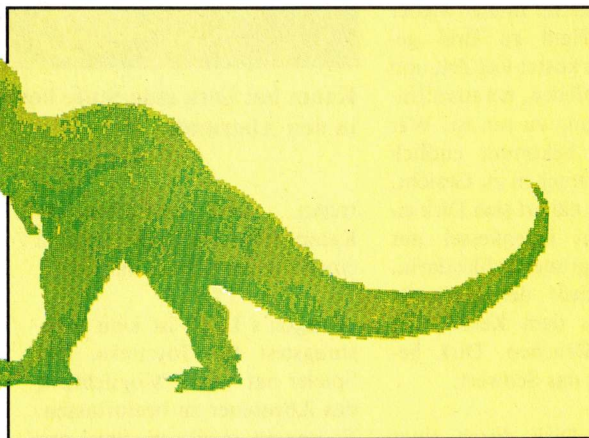


Bilder malen, verändern, beleuchten, bewegen und konvertieren – AmigaWelt zeigt wie man's macht und welche Ausstattung notwendig ist. Außerdem werden effektvolle Programmanwendungen vorgestellt. Erscheinungstermin: 19.4.89



Multitalent in Video

„Video Toaster“ kann fast alles. Er taugt zwar nicht zum Brotrösten, aber zum Digitalisieren, Genlocken, Zoomen, Verbiegen, Verzerren, Spiegeln und Flipping von Bildern — und all das in Echtzeit. Zunächst wird das Multitalent von NewTek als NTSC-Version getestet.



Tasks im Verbund

Der mächtige Tyrannosaurus Rex hat einen neuzeitlichen Nachfolger erhalten: ARexx, eine neue Makro-Hochsprache, herrscht seit kurzem als gewaltiges Urwesen über die Multitasking-Umgebung des Amiga. Programme wie AmigaTEX, TxED, VLT-Network oder der Terminmanager NAG lassen sich durch ständigen Datenkontakt zu neuen Programmgebilden zusammenfassen.

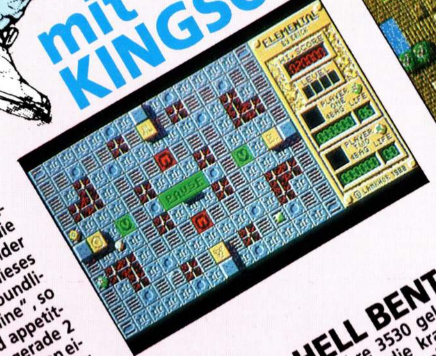


45 Stunden Klon-Suche

Spieleprogrammierer Blood ist durch einen Fehler in sein eigenes Spiel geraten. Und nicht nur das — er ist 30mal geklont worden. Überall im Universum befinden sich die Kopien seiner selbst, und jede zehrt an Bloods Lebensenergie. „Die Arche des Captain Blood“ führt den Spieler durch den Weltraum auf der Suche nach Blood-Kopien — 45 Stunden lang.



mit KINGSOFT-Spielen:





ELEMENTAL

Elemental ist ein ganz besonderer Planet. Es ist eine viereckige Welt, in der die Elemente nicht wirklich miteinander verbunden sind. Die Lebewesen dieses Planeten ernähren sich von "Roundlines" ... und Sie sind ein "Roundline", so wie er sein sollte, d.h., Sie sind appetitanregend. Außerdem sind Sie gerade 2 Jahre alt. Ein wichtiges Alter im Leben eines "Roundline". In dieser Zeit muß er die große Prüfung bestehen, vor der alle jungen "Roundlines" vor Angst zittern. Sind Sie in der Lage, die 32 Stationen der Prüfung zu überstehen?

Atari ST

79,95

GALAXY'89

Eine Neuauflage des berühmten Klassikers. Natürlich mit allen bekannten Features wie Mehrfach-Raumschiff, Bounsrunde, usw., aber jetzt mit zeitgemäßer Amiga-Grafik und Digi-Sounds.

Amiga

39,95



HELL BENT

Im Jahre 3530 gelang es Captain Drak Hellbent, die kraellischen Thronräuber aus ganz Aldonlcha zu vertreiben. Erleben Sie mit dieser Computer-Präsentation eine Legende mit viel Action, in einer zerklüfteten, gewundenen Landschaft, die Ihnen durch HELLBENT nähergebracht wird.

69,95

Atari ST, Amiga



GRAVITY-FORCE

Erleben Sie die Schwerkraft so realistisch wie nie zuvor. Mit Ihrem Raumtransporter müssen Sie Ladung aufnehmen, sich durch feindliche Linien durchkämpfen und im Hauptquartier abliefern. Zusätzlich werden Ihnen weitere Spielarten geboten., z.B. 2 Spieler-Simultan-Wettrennen oder 2 Spieler im Kampf gegeneinander.

39,95

Amiga



EVOLUTION CRYSER

Wir schreiben das Jahr 2014. Der Kommandant der Raumflotte Alpha 1 gibt Befehl, feindliche Rebellen, die im Quadranten Tetron ihr Unwesen treiben, aufzuhalten. Sein bester Mann, der Pilot Farrow, macht sich zu dieser Mission bereit. Zur Verfügung steht ihm ein Raumschiff, das er je nach Geschick mit verschiedenen Extrawaffen ausrüsten kann - wird er es schaffen?

39,95

Amiga



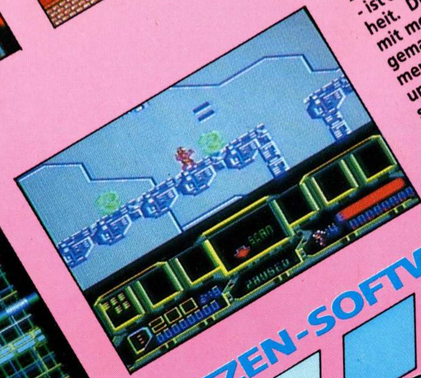
OVERFLOW

Ein Strategiespiel für alle Tüftler und Knobler. Wer schafft es zuerst, seinen Gegner (oder den Computer) vom Feld zu drängen? Vorsicht: Dieses Spiel kann schnell süchtig machen!

Amiga

29,95





HOTBALL

Das lange Warten hat sich gelohnt: mit Hotball steht jetzt die erste wirklich gute Fußball-Simulation für 16-Bit-Computer zur Verfügung. Super-Grafik, enormer Spielspaß und der neuartige 4-Spieler-Modus (gleichzeitig!) machen Hotball zum Suchtspiel. Lieferung inklusive Adapter, um 2 weitere Joysticks an Ihren Computer anzuschließen.

79,95

Atari ST, Amiga (1MB)

X-TERMINATOR

- ist der Deckname für ein Mitglied einer Elite-Sonder-Einheit. Diese ausgebildeten Einzel-Kämpfer, ausgerüstet mit modernster militärischer Technologie, wurden mobil gemacht, um die "Bio-Morph"-Bedrohung, die die menschlichen Kolonien heimsucht, ausfindig zu machen und zu vernichten. X-TERMINATOR - das sind 15 anspruchsvolle und herausfordernde Levels der strategisch taktischen Action!

C-64 Kassette 34,95

C-64 Diskette 49,95

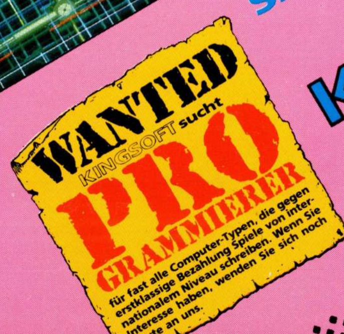


HARD LINES

Ein spannender Wettkampf mit tödlichen Linien für 1 oder 2 Spieler gegeneinander oder gegen den Computer. Viele schöne Hintergrundgrafiken, Digisounds und zahlreiche Spielvariationen mit geteiltem, scrollenden Bildschirm werden Sie lange an Ihren Computer fesseln.

29,95

Amiga



SPITZEN-SOFTWARE

KINGSOFT

MADE IN GERMANY

Grüner Weg 29
5100 Aachen
Tel. 0241/15 20 51
Fax 0241/15 20 54

Dies ist nur ein kleiner Ausschnitt aus unserem riesigen Softwareangebot. Fordern Sie bitte noch heute kostenlose Informationen an.

Alle Spiele selbstverständlich mit ausführlicher deutscher Anleitung. Bitte bei Ihrer Bestellung unbedingt den Computer-Typ angeben.